

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab, Blida
USDB.

Faculté des sciences.
Département informatique.



**Mémoire pour l'obtention
du diplôme d'ingénieur d'état en informatique.**
Option : Intelligence Artificielle

Sujet :

Conception et réalisation d'une
application de commerce
électronique à base d'agents mobiles
pour la recherche et la vente des
formations sur Internet

Présenté par : Mlle AISSATI Safia
Mlle FODIL CHERIF Fouzia

Encadreur : Mr D.E MENACER

Organisme d'accueil : LOGITEL – CTEC MICROSOFT

Soutenus le: 16/10/2007, devant le jury composé de :

Mme Benstiti,
Mme Aoussat,
Mr Bala.

Président
Examineur
Examineur

2006-2007

MIG-004-164-1

Remerciements

Au seuil de ce travail, nous tenons à remercier profondément notre encadreur, monsieur DJAMEL EDDINE MENACER, qui, sans sa précieuse assistance, ce mémoire n'aurait sans doute pas abouti.

Nous tenons également à remercier tous les enseignants de l'université de Blida au département de l'Informatique, y compris monsieur le chef de département pour nous avoir offert l'opportunité d'effectuer notre stage.

Nos remerciements les plus distingués sont adressés aux membres du jury qui nous ont fait l'honneur de bien vouloir accepter d'évaluer ce modeste travail.

La liste n'est pas exhaustive: Nous remercions toutes les personnes, des amis ou inconnus, qui, même au cours d'une rencontre fugace, nous ont fait découvrir le plaisir d'apprendre.

Dédicace

*A ma raison de vivre, d'espérer,
à ma source de courage, à ceux que j'ai de plus cher:
mes parents,
pour leur amour, leur confiance et leur sacrifice sans limite,
mes trois adorables frères:
A. Errahmane, A. Elwahab, et Djihed.
A toute ma famille, en particulier à ma grande mère,
A mon binôme et ma chère amie : Fouzia
A mes très chères amies :
Rima, Houria, Baya, Hassiba, Samia, Imen, Assia, Nawel, Amel,
Salma, Souad, Mahjouba, ...ect,
A Allal, Billel, Mounir, Samir, Youcef,
A tous mes collègues d'études.*

AISSATI Safia

Dédicace

A ma raison de vivre, d'espérer,

à ma source de courage, à celle que j'ai de plus cher:

ma maman,

pour son amour, son confiance et son sacrifice sans limite,

A l'âme de mon père,

A mon adorable frère:

Amine,

A mon bijou, ma très très chère Mignonne, et raisonnable petite jolie ange :

Hadil

A toute ma famille, en particulier à ma tante Houria, et Adem,

A mon binôme et ma chère amie : Safia

A mes très chers amies :

Rima, Houria, F.Zohra, Baya, Hassiba, Nadia, Nawel, Mahjouba,

Amitaf, Samia, Hiba, Chafika, Imene,...ect,

A Mohamed, Allal, Billel, Mounir, Samir, Hamza, Madjid, Youcef,

A tous mes collègues d'études.

FODIL CHERIF Fouzia

Table des matières

Résumé.....	1
Introduction générale.....	2
CHAPITRE I : LA TECHNOLOGIE DES AGENTS	
A. Agent et système multi-agents	4
1. INTRODUCTION	4
2. DEFINITION DE L'AGENT	4
3. TYPE D'AGENTS	5
3.1. Les agents réactifs	5
3.2. Les agents cognitifs	5
3.3. Comparaison entre les deux types d'agents	5
4. CARACTERISTIQUES DES AGENTS	5
5. LES AGENTS INTELLIGENTS	6
5.1. Définition	6
5.2. Composants de l'agent intelligent	7
5.3. Domaines d'application de l'agent intelligent	7
5.3.1. Agent d'interface	8
5.3.2. Agent de recherche d'information	8
5.3.3. Agent commercial	8
6. LANGAGE DE PROGRAMMATION DES AGENTS	8
6.1. Java	8
6.2. LISP	8
6.3. C /C++	9
6.4. Python	9
6.5. PHP	9
7. LES SYSTEMES MULTI-AGENTS	9
7.1. Définition	9
7.2. Architecture d'un système multi-agents	9

7.3. Communication entre les agents	10
7.3.1. Communication asynchrone	10
7.3.2. Communication synchrone	11
• La coopération	11
• La négociation	11
• L'optimisation de ressources	11
• Langages de communication entre agent.....	11
- KIF	12
- KQML	12
- FIPA-ACL	12
8. CONCLUSION.....	12
B. Les agent mobiles	13
1. INTRODUCTION.....	13
2. DEFINITION D'UN CODE MOBILE.....	13
3. DEFINITION D'UN AGENT MOBILE.....	13
4. LE CYCLE DE VIE D'UN AGENT MOBILE	13
5. CARACTERISTIQUE D'UN AGENT MOBILE	14
5.1. La mobilité.....	14
5.2. Réduction de la charge dans le réseau.....	14
5.3. Exécution asynchrone et autonome.....	15
5.4. Adaptation dynamique à l'environnement	15
5.5. Robustesse et tolérance aux fautes.....	16
5.6. Communication.....	16
6. LA MIGRATION.....	16
6.1. La migration forte	17
6.2. La migration faible	17
7. FONCTIONNEMENT.....	17
8. LA SECURITE DANS LES SYSTEMES D'AGENTS MOBILES.....	17
8.1. La sécurité de l'agent visiteur	17

- La sécurité de l'hôte visité	17
- L'infrastructure de communication	17
- Sécurité entre deux agents.....	18
9. QUELQUES SYSTEMES D'AGENT MOBILES EXISTANTS	18
9.1. Le système Telescript.....	18
9.2. Le système Aglets.....	19
9.3. Agent-Tcl	19
9.4. TACOMA	20
9.5. CONCORDIA.....	21
9.6. Voyager.....	22
10. JAVA : Langage De Programations Des Agents Mobiles.....	23
11. AVANTAGES DES AGENTS MOBILES.....	23
12. CONCLUSION.....	27
CHAPITRE II : L'E-COMMERCE ET LES AGENTS	
A. Le commerce électronique.....	28
1. INTRODUCTION	28
2. A Quoi Sert Le Commerce Electronique?	28
3. DEBUT DE COMMERCE ELECTRONIQUE.....	28
4. DEFINITION DE L'E-COMMERCE.....	29
5. TYPOLOGIES DE COMMERCE ÉLECTRONIQUE	29
5.1. Business to Business (B2B)	30
5.2. Business to Customer (B2C)	30
5.3. Consumer to Business (CtoB)	30
5.4. Consumer to Consumer (CtoC)	30
5.5. Business to Government (BtoG)	30
5.6. Government to Consumer (GtoC)	30
5.7. Employe to Employe (EtoE)	30
6. LES DIFFERENTES TACHES DE E-COMMERCE.....	30

7. COMPARAISON ENTRE LE COMMERCE ELECTRONIQUE ET LE COMMERCE TRADITIONNEL.....	31
8. L'ACHAT EN LIGNE.....	32
9. PLACE DE MARCHE (MARKET PLACE)	33
10. CARACTERISTIQUE DU COMMERCE ELECTRONIQUE	33
11. LES PROBLEMES DE LA SECURITE DES TRANSACTIONS SUR INTERNET.....	33
11.1. La confidentialité	33
11.2. L'authenticité	34
11.3. Non-répudiation	34
11.4. L'intégrité	35
11.5. Contrôle de l'accès	35
11.6. Disponibilité.....	34
12. MODES DE PAIEMENTS.....	34
12.1. Crédit.....	35
12.2. Porte-monnaie électronique.....	35
13. MODES DE SECURISATION DE PAIEMENT	35
13.1. SSL (Secure Socket Layer)	35
13.2. SET (Secure Electronic Transaction).....	35
13.3. C-SET (Chip Secure Electronic Transaction)	36
14. DEFINITION DE NEGOCIATION	36
15. AVANTAGES DU COMMERCE ELECTRONIQUE.....	36
16. AVENIR DE COMMERCE ELECTRONIQUE.....	37
17. CONCLUSION.....	37
B. L'utilisation des agents dans le commerce électronique.....	38
1. INTRODUCTION	38
2. POURQUOI LE COMMERCE ELECTRONIQUE UTILISE LES AGENTS ?	38
3. LES BENEFICES D'UTILISATION DES AGENTS DANS L'E-COMMERCE.....	38
4. TYPES D'AGENTS DANS LE COMMERCE ELECTRONIQUE.....	38
4.1. Les agents acheteurs.....	39

4.2. Les agents vendeurs	39
5. LA NEGOCIATION ENTRE AGENTS.....	40
6. LE COMPORTEMENT D'ACHAT	40
6.1. Identification des besoins	40
6.2. Courtage de produit (product brokeing)	40
6.3. Courtage de marchands (merchant brokering)	40
6.4. Négociation.....	41
6.5. Achat et livraison.....	41
6.6. Service après vente et évaluation.....	41
7. APPLICATIONS RECENTES D'AGENTS INTELLIGENTS EN COMMERCE ELECTRONIQUE	41
7.1. Les agents du courtage de produits.....	41
7.2. Les agents du courtage des marchands.....	42
7.3. Les agents de négociation	42
8. QUELQUES ARCHITECTURES DES SYSTEMES DE COMMERCE ELECTRONIQUE EXISTANTES A BASE D'AGENTS MOBILE.....	43
8.1. Kasbah.....	43
8.2. MAGMA.....	44
8.3. MarketSpace.....	45
9. CONCLUSION	46
CHAPITRE III : PROPOSITION D'UNE SOLUTION AU COMMERCE ELECTRONIQUE SUR INTERNET	
1. INTRODUCTION.....	47
2. PROBLEMATIQUE.....	47
2.1. Protection d'un site hôte contre un agent	47
2.2. Protection d'un agent contre un site hôte.....	48
3. PROPOSITION.....	49
4. ARCHITECTURE PROPOSEE.....	51
3.1. Type de sites.....	52
3.2. Type d'agent.....	52

3.3. Composants de l'architecture	53
4. REPRESENTATION PHYSIQUE D'UNE MP.....	54
5. LES SERVICES S'EXECUTANT DANS UNE PLACE CE MARCHE.....	55
6. BASES DE CONNAISSANCES.....	55
6.1. La base MPNS	55
6.2. La base d'annuaire.....	56
6.3. La base d'e-shop.....	56
6.4. Flux de données entre les différentes bases de connaissances.....	56
7. LES ENVIRONNEMENTS ASP (AGENT SERVICES PROVIDER)	57
8. LA SECURITE DU SYSTEME.....	57
8.1. Les problèmes posés.....	57
8.2. Les solutions possibles.....	57
9. CONCLUSION.....	60
CHAPITRE IV : CONCEPTION DU SYSTEME	
1. INTRODUCTION.....	61
2. DESCRIPTION DU MODELE DE COMMERCE ELECTRONIQUE.....	61
2.1. Rechercher une formation.....	63
2.2. Négociation.....	63
2.3. commander une formation.....	64
2.4. Offrir une formation.....	64
2.5. Diagramme de cas d'utilisation détaillé de modèle de commerce électronique « Formation en ligne ».....	65
2.6. Diagramme de classe de modèle de commerce électronique.....	67
3. DESCRIPTION DU MODELE DE SYSTEME MPSMA.....	68
3.1. Diagramme de cas d'utilisation	68
3.1.1. Commande une formation	69
3.1.2. Offre une formation.....	71
3.2 SCENARIOS ET DIAGRAMME DE SEQUENCES.....	72
4. DIAGRAMME DE CLASSE	92
5. CONCLUSION.....	93

CHAPITRE V : IMPLEMENTATION ET TESTS

1. INTRODUCTION.....	94
2. CHOIX DU LANGAGE DE PROGRAMMATION.....	94
3. REALISATION DU SYSTEME MPSMA.....	96
3.1. Implémentation.....	96
3.1.1. Implémentation de la création des agents.....	96
3.1.2. Implémentation de la mobilité des agents	97
3.1.3. Implémentation de la communication inter-agents.....	99
3.1.4. Implémentation de la sécurité des agents.....	100
3.2. Implémentation de l'exécution des agents.....	100
3.2.1 Implémentation de l'agent acheteur.....	100
3.2.2. Implémentation de l'agent vendeur.....	102
3.2.3. Implémentation de l'agent facilitateur du MP.....	103
4. REALISATION DE L'APPLICATION	106
4.1. Réalisation de la commande d'une formation en ligne.....	106
4.2. Réalisation de l'offre d'une formation en ligne.....	111
5. TEST.....	111
6. CONCLUSION	117
CONCLUSION GENERALE.....	118
OBJECTIFS.....	119
PERSPECTIVE.....	120
ANNEXES.....	121
Annexe A. KQML.....	121
Annexe B. RSA.....	123
Annexe C. UML.....	124
BIBLIOGRAPHIE.....	127
WEB GRAPHIE.....	121

Liste des tableaux

Tableau I.1 : Comparaison entre les agents réactifs et les agents cognitifs.....	5
Tableau I.2 comparaison entre les modèles client/serveur et agents mobile.....	16
Tableau II.1 Les différentes phases en e-commerce.....	31
Tableau IV.1 Exemple des éditeurs MP.....	68
Tableau IV.2 Exemple de spécialité d'e-shop.....	69
Tableau V.1 : Exemple	112
Tableau V.2 : Exemple sur l'enregistrement de la trace d'une opération d'achat	117

Liste des figures

Figure I.1: Représentation de l'intelligence d'un Agent.....	6
Figure I.2 : Composants de l'agent intelligent.....	7
Figure I.3 Architecture d'un SMA à contrôle centralisé.....	10
Figure I.4 Architecture d'un SMA à contrôle distribué.....	10
Figure I.5 Le paradigme client/serveur.....	14
Figure I.6 Le paradigme des agents mobiles.....	14
Figure I.7 Architecture Client/Serveur.....	15
Figure I.8 L'asynchronisme des agents mobiles.....	15
Figure I.9 Le système Telescript.....	18
Figure I.10 Le cycle de vie d'un agent Agent-Tcl.....	19
Figure I.11 Les Folders de TACOMA.....	21
Figure I.12 Rendez-vous distant de TACOMA.....	21
Figure II.1 L'architecture de MAGMA.....	45
Figure III.1 Interaction des agents mobiles à travers un proxy.....	50
Figure III.2 Interaction des agents mobiles à travers une place de marché.....	50
Figure III.3 Différentes implémentation du schéma SB.....	51
Figure III.4 Architecture proposée.....	52
Figure III.5 Environnement Internet pour notre architecture.....	53
Figure III.6 Schéma physique d'une place de marché.....	54
Figure III.7 schéma logique d'une place de marché.....	55
Figure III.8 Différentes bases de connaissances et flux correspondants.....	56
Figure III.9 vue générale de notre architecture proposée.....	59
Figure IV.1 diagramme de cas d'utilisation générale d'application de commerce électronique.....	62
Figure IV.2 diagramme de cas d'utilisation de <i>Commande une formation</i>	62
Figure IV.3 diagramme de cas d'utilisation d'Identification des besoins.....	63
Figure IV.4 diagramme de cas d'utilisation de recherche une formation.....	64

Figure IV.5 diagramme de cas d'utilisation de validation une formation.....	65
Figure IV.5 diagramme de cas d'utilisation d'offrir une formation.....	67
Figure IV.6 diagramme de cas d'utilisation détaillé de modèle de commerce électronique.....	66
Figure IV.7 Décomposition de fonction commande une formation en sous fonctions.....	70
Figure IV.8 Décomposition de fonction d'offrir une formation en sous fonctions.....	71
Figure IV.9 diagramme de séquence de l'inscription d'un client.....	71
Figure IV.10 diagramme de séquence de création de BMA.....	73
Figure IV.11 diagramme de séquence de création d'un compte VAN.....	73
Figure IV.12 diagramme de séquence de certification de BMA.....	74
Figure IV.13 diagramme de séquence de migration de BMA vers les MPs.....	76
Figure IV.14 diagramme de séquence de migration de BMA vers les e-shops.....	78
Figure IV.15 diagramme de séquence de négociation.....	79
Figure IV.16 diagramme de séquence de la sécurité de transaction.....	80
Figure IV.17 diagramme de séquence de paiement.....	81
Figure IV.18 diagramme de séquence de Les mises à jour de bases de connaissances.....	82
Figure IV.19 Diagramme de séquence de retour de BMA.....	82
Figure IV. 20 Diagramme de séquence de l'inscription du fournisseur.....	83
Figure IV.21 diagramme de séquence de création de SMA.....	84
Figure IV.22 diagramme de séquence de certification de SMA.....	85
Figure IV.23 Diagramme de séquence de migration de SMA vers les MPs.....	87
Figure IV.24 diagramme de séquence de migration de BMA vers les e-shops.....	89
Figure IV.25 diagramme de séquence de négociation.....	90
Figure IV.26 Diagramme de séquence de la sécurité d'interaction.....	91
Figure IV.27 Diagramme de séquence de Les mises à jour de bases de connaissances.....	92
Figure IV.28 Diagramme de séquence de retour de SMA.....	92
Figure V.1 La forme de la class agent.....	97
Figure V.2 La forme de la class agent mobile.....	99
Figure V.3 forme de l'interface message.....	100
Figure V.4 forme de la class agent acheteur.....	101
Figure V.5 forme de la class agent vendeur.....	103
Figure V.6 La forme de la class agent facilitateur.....	105
Figure V.7 Index du site.....	107
Figure V.8 Exemple de description d'une formation sélectionnée.....	108
Figure V.9: formulaire d'inscription.....	109

Figure V.10 : spécification des besoins d'un client.....	110
Figure V.11 : bon de commande.....	111
Figure V.12 Exemple d'identification d'client.....	113
Figure V.13 Exemple d'identification des besoins d'client.....	114
Figure V.15 Exemple d'un résultat apporté à un client.....	115
Figure V.16 Exemple d'une étape de paiement effectuée par à un client.....	116

Résumé

On parle d'agents mobiles depuis des années, mais la technologie des agents reste plutôt pénible à mettre en œuvre. La *sécurité* reste le problème le plus difficile à résoudre pour l'utilisation des agents mobiles. C'est aussi la raison qui empêche leur diffusion.

Comme tout système distribué, les agents mobiles sont sujets aux menaces du réseau comme : la corruption de données, le masquerading, le déni de service, les écoutes, la répudiation... etc.

Les solutions à ces problèmes existent : chiffrement, autorisation, authentification, non-répudiation, etc. Mais ces solutions sont adaptées aux systèmes répartis classiques (par exemple : client-serveur).

Les recherches actuelles n'offrent pas de méthode générale pour protéger les agents, mais seulement pour protéger les hôtes visités. Les développeurs d'applications à agents mobiles doivent concevoir leurs propres mécanismes de protection selon leurs besoins.

Nous nous proposons d'utiliser un environnement pour développer un système de e-commerce à base d'agents mobiles sécurisés.

Mots clés: Agents mobiles, applications réparties, sécurité, cryptage, e-commerce, Java.

Abstract

We speak for years of mobile agents, but the technology of the agents remained rather laborious to use. The security remains the most difficult problem to solve for the use of the mobile agents. It is also the reason that prevents their diffusion.

As all distributed system, the mobile agents are subject to the threats of the network as: the corruption of data, the masquerading, the denial of service, monitoring, the répudiation.etc.,.

The solutions to these problems exist: cipherring, authorization, authentication, non repudiation, etc. But these solutions are adapted to the classic distributed systems (for example: client-server).

The present researches don't offer a general method to protect the agents, but only to protect the visited hosts. The developers of applications to mobile agents must conceive their own protective mechanisms according to their needs.

We intend to use an environment to develop an e-commerce system to basis of secured mobile agents.

Keywords

Mobile agents, distributed applications, security, encryption, e-commerce, Java,

Introduction générale

Avec le paradigme client-serveur, l'espace de l'informatique décisionnel s'est créé et grâce à la technologie de l'Internet, l'ère de l'informatique pour la communication s'ouvre.

Nombreuse services et applications sur Internet fonctionnent selon le paradigme client-serveur, dont nous trouvons les applications de commerce électronique qui occupent aujourd'hui une place plus prépondérante dans l'univers des télécommunications.

Toutefois, ces applications possèdent de gros inconvénients, parmi lesquels nous trouvons :

- La difficulté à gérer la sécurité sur Internet.
- Le problème du temps de réponse.

L'autonomie, l'asynchronisme, l'intelligence et la mobilité d'un agent permettent de diminuer le temps de réponse, de plus ces caractéristiques permettent aux utilisateurs de trouver un meilleur produit au meilleur prix en moindre coût.

Les notions de sécurité des agents que nous considérons sont la confidentialité et l'intégrité. La confidentialité porte sur la non-divulgence des informations alors que l'intégrité porte sur leur non-modification.

Notre travail consiste à créer un site web de commerce électronique, en utilisant la technologie d'agent mobile et les notions de sécurité.

A fin de présenter le but de notre travail, nous avons organisé notre mémoire comme suit :

- Chapitre I se compose de deux parties :

✓ 1^{ère} Partie : nous présentons la technologie des agents et les agents intelligents, aussi nous présentons les systèmes multi-agents.

✓ 2^{ème} Partie : nous présentons la technologie d'agent mobile, en la comparant avec le système client-serveur. Nous passons ensuite aux principaux systèmes à base d'agents mobiles.

• Chapitre II se compose aussi de deux parties :

✓ 1^{ère} Partie : nous présentons le commerce électronique et ensuite les problèmes de sécurité, les modes de paiement et leurs modes de sécurisation.

✓ 2^{ème} Partie: nous présentons comment et pourquoi le commerce électronique utilise les agents, nous présentons ensuite quelques systèmes de commerce électronique à base d'agents mobiles.

• Chapitre III : nous posons une problématique liée au commerce électronique, ensuite nous proposons une architecture comme une solution au commerce électronique sur Internet avec les détails de tous ses aspects.

• Chapitre IV : avec l'utilisation du langage de modélisation UML, nous faisons une conception du système en détaillant les différentes fonctionnalités que le système doit offrir aux usagés.

• Chapitre V : nous faisons une implémentation de la solution retenue au niveau de la conception.

Nous achevons notre mémoire par une conclusion et perspectives

Chapitre I

La technologie des agents

A- Agent et système multi-agents

1. INTRODUCTION

Depuis 1993 et le démarrage du World Wide Web, le nombre de sites, le nombre de pages, le nombre d'utilisateurs augmentent sans arrêt. Mais quelques problèmes sont apparus, les temps de réponse catastrophiques, les recherches difficiles dans cette montagne d'information, la sécurité des sites, la confidentialité des informations, la criminalité naissante, la protection de l'ensemble du réseau et son administration. Pour diminuer ces problèmes la technologie des Agents arrive.

2. DEFINITION DE L'AGENT

Le domaine des Agents est très récent. Et comme dans le cas de toute technologie nouvelle il n'y a pas de définition universelle. Nous citons, par exemple, les définitions suivantes :

- **Définition « KidSim »**

Un agent est une entité logicielle persistante dédiée à une tâche spécifique.

- **Définition « IBM »**

Entité logicielle assistant les utilisateurs et agissant pour leur compte.

- **Autre définition**

Un agent est une entité physique ou virtuelle qui est capable d'agir dans son environnement, communiquer avec d'autres agents, doué d'autonomie, possède des ressources propres, capable de percevoir son environnement et de s'adapter à ses modifications, et peut éventuellement se reproduire. [1,2]

3. TYPES D'AGENT

3.1. Les agents réactifs

Ces agents ne possèdent pas une représentation complète de son environnement et ne sont pas capable de tenir compte de ses actions passées. Ils ont un comportement du type « stimulus – réponse ». [3,4]

3.2. Les agents cognitifs

Ces agents sont plus évolués. Ils ont une représentation globale de leur environnement et des autres agents avec lesquels ils communiquent. Ils savent tenir compte de leur passé et s'organisent autour d'un mode social d'organisation. [3,4]

3.3 Comparaison entre les deux types d'agents

La différence entre les deux types d'agents est représentée dans le tableau ci-dessous :

Tableau I.1 : Comparaison entre les agents réactifs et les agents cognitifs [5].

Agents cognitifs	Agents réactifs
-Représentation explicite de l'environnement	-Pas de représentation explicite
-Peut tenir compte de son passé	-Pas de mémoire de son histoire
-Agent complexe	-Fonctionnement stimulus/ action
-Petit nombre d'agents	-Grand nombre d'agents

4. CARACTÉRISTIQUE DES AGENTS

On peut identifier [6, 7, NIES 02] les caractéristiques suivantes d'un agent:

- **Situé** : l'agent est capable d'agir sur son environnement à partir des entrées sensorielles qu'il reçoit de ce même environnement.
- **Autonome**: l'agent est capable d'agir sans l'intervention d'un tiers (humain ou agent) et contrôle ses propres actions ainsi que son état interne.

- **Proactif**: l'agent doit exhiber un comportement proactif et opportuniste, tout en étant capable de prendre l'initiative au bon moment.
- **Capable de répondre à temps** : l'agent doit être capable de percevoir son environnement et d'élaborer une réponse dans le temps requis.
- **Social** : l'agent doit être capable d'interagir avec des autres agents (logiciels ou humains) afin d'accomplir des tâches ou aider ces agents à accomplir les leurs.

5. LES AGENTS INTELLIGENTS

5.1. Définition

Un agent intelligent est un programme qui assiste et exécute des travaux qui lui ont été délégués au nom de l'utilisateur. Il est autonome, mobile, social, animé par un objectif, réactif et s'adapte à son environnement. Et en plus il possède la capacité de raisonner et d'apprendre, ainsi un certain degré d'intelligence. [9,10]

L'intelligence était basée sur un agrégat de caractéristiques (la capacité d'apprendre, la capacité sociale et une haute autonomie) chacun pouvant avoir des degrés divers.

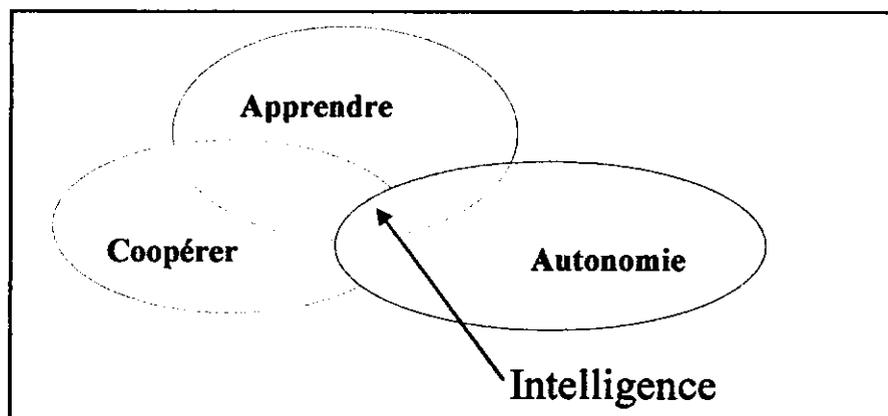


Figure I.1: Représentation de l'intelligence d'un Agent [9,10].

5.2. Composants de l'agent intelligent

Un agent intelligent [8, 11,12] contient un ou plusieurs éléments suivants :

- **Le moteur «le cerveau de l'agent »** : lui permet de tenir des raisonnements plus ou moins complexes.
- **Une base de connaissance prédéfinie** : ce que l'agent sait, croit et pense. La connaissance peut être basée sur des applications existantes.
- **L'interface avec les autres applications** : C'est l'interface qui détermine le champ d'application d'un agent.
- **L'interface avec l'utilisateur** : permet la communication entre l'agent et l'utilisateur. L'utilisateur donne ses instructions à l'agent et ce dernier renvoie les résultats ou donne des informations.
- **Un mécanisme d'apprentissage.**

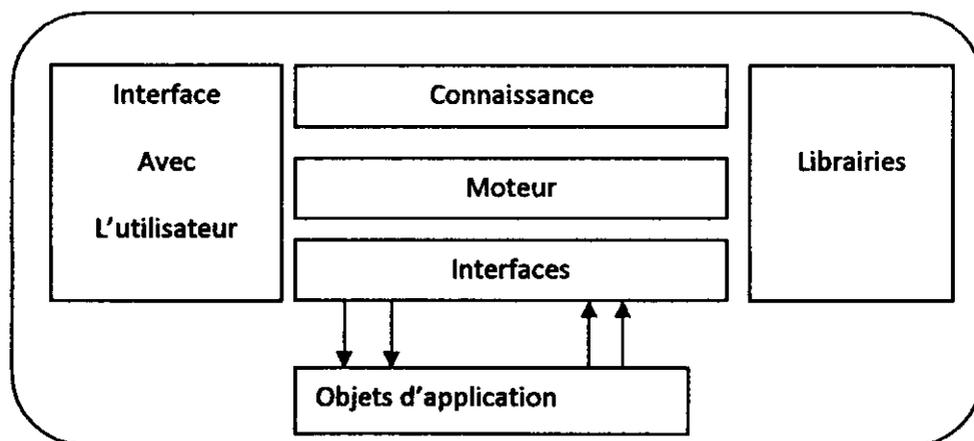


Figure I.2 : Composants de l'agent intelligent [8,12].

5.3. Domaines d'application de l'agent intelligent

Les domaines d'application [17] des agents intelligents sont loin d'être limités. Il est cependant possible de distinguer :

5.3.1. Agent d'interface

L'agent d'interface observe les actions de l'utilisateur, lui suggère des astuces, lui apprend des raccourcis.

5.3.2. Agent de recherche d'information

Un agent de recherche d'information est un agent qui a accès à une ou plusieurs sources d'information et qui est capable de récolter et manipuler cette information pour répondre à des demandes d'utilisateurs ou d'autres agents.

5.3.3. Agent commercial

Son rôle est de faciliter aux consommateurs la sélection de boutiques, de marques ou de produits en se basant sur leurs goûts et permettre aux vendeurs de mieux connaître la demande.

6. LANGAGES DE PROGRAMATION DES AGENTS

Les différents langages de programmation utilisés dans la conception des Agents :

6.1. Java

Un agent a besoin d'une machine virtuelle pour fonctionner et Java en possède une. De plus le langage Java est sécurisé car c'est la JVM qui vérifie au moment de l'exécution la fiabilité du code ; et aussi Java est orienté objet et tout ceci confère à Java une bonne place pour la construction d'agents. [SAGG 01]

6.2. LISP

C'est un langage [13] fonctionnel très puissant et flexible, il est utilisé en intelligence artificielle. Il sera utile pour implémenter les mécanismes de déduction de l'agent.

6.3. C /C++

Sont des langages [SAGG 01,13] qui ont une très grande puissance et une vitesse d'exécution inégalée. Mais ils ne sont pas sécurisés, ce qui peut entraîner un « plantage » des machines qui exécuteraient un processus agent basé sur ces langages.

6.4. Python

Python [SAGG 01] offre de nombreux avantages tels que la possibilité d'être compilé ou interprété et d'être orienté objet. De plus, il a toutes les caractéristiques et la puissance d'un langage pouvant créer des Agents.

6.5. PHP

Est un langage [SAGG 01] script s'exécutant sur le serveur, comparable dans ces grandes lignes au JavaScript serveur. Il permet notamment d'interagir entre des SGBDR situés sur un serveur et un poste client équipé d'un navigateur. Cela lui donne de grandes capacités pour la création d'agents.

7. LES SYSTÈMES MULTI-AGENTS

7.1. Définition

Un SMA [3,14] est un ensemble d'agents situés dans un certain environnement, interagissant selon une certaine organisation.

7.2. Architecture d'un système multi-agents

➤ Dans un SMA à contrôle centralisé ou à base de tableau noir, l'interaction entre agents se fait via le partage d'un même espace de travail avec l'absence de communication directe entre eux [5]. Autrement dit, un agent connaît tous les autres agents. L'avantage de cette méthode est sa simplicité de mise en œuvre. Cependant, un seul objet gérant toutes les adresses, il peut devenir un goulet d'étranglement et grever de façon significative les performances du système. De même, si l'objet gérant le système d'adressage plante, le système plante avec [OLIV 02].

Un SMA à contrôle centralisé est composé de: [5]

- Les connaissances.
- Un tableau noir.
- Un mécanisme de contrôle.

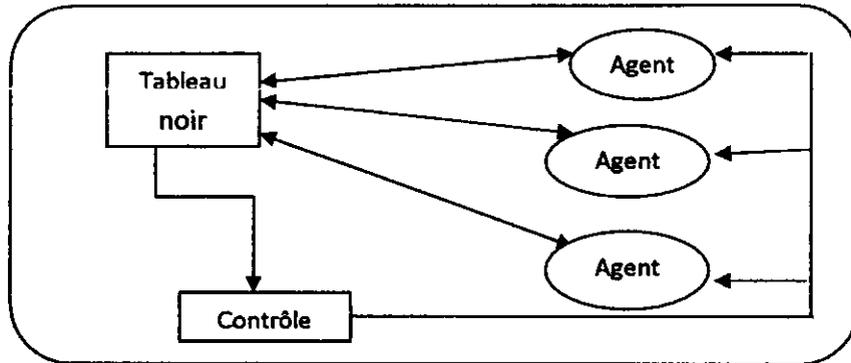


Figure I.3 Architecture d'un SMA à contrôle centralisé. [5]

➤ Dans un SMA à contrôle distribué la communication entre agents se fait par envoi de message, ainsi il a une distribution totale des connaissances et du contrôle. [05] Donc personne n'a, a priori, de vision globale. En cas de défaillance d'une partie du système, le reste peut continuer à fonctionner. [OLIV 02]

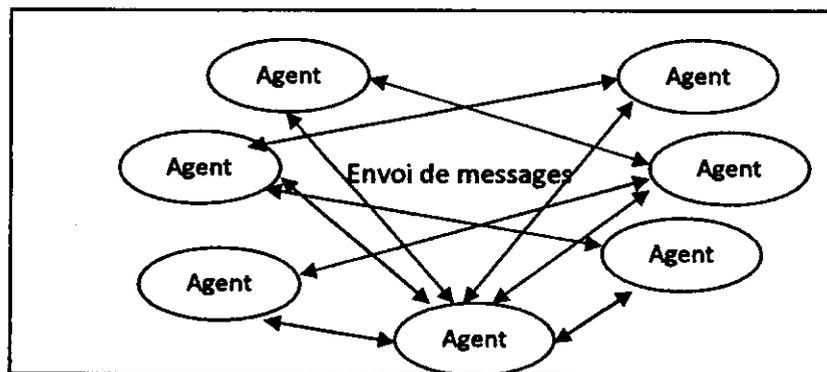


Figure I.4 Architecture d'un SMA à contrôle distribué. [5]

7.3. Communication entre les agents

7.3.1. Communication asynchrone

La réponse non donnée immédiatement. On demande et on n'attend pas la réponse, on fait autre chose en attendant [OLIV 02].

7.3.2. Communication synchrone

Dans la communication synchrone un objet demande quelque chose à un autre objet et attend la réponse avant de poursuivre son exécution [OLIV 02].

Dans les deux types de communications nous distinguons :

- **La coopération**

La coopération s'effectue quand les agents ont un but en commun, et leurs actions tendent à réaliser ce but.

- **La négociation**

La négociation correspond [15] à la coordination entre agents. Dans les systèmes multiagents, l'objectif est de construire des agents négociants avec intervention humaine la plus réduite possible. Par exemple dans le commerce électronique, l'agent acheteur et l'agent vendeur négocient le prix d'un bien ou d'un service

- **L'optimisation de ressources**

L'optimisation de ressources [15] concerne le développement de stratégies de négociation optimales. Par exemple, dans le e-commerce un agent acheteur peut perdre une utilité avec le temps comme résultat de ne pas avoir le service. D'autre part, un agent vendeur peut gagner plus d'utilité en offrant le service le plus tardivement possible. Ainsi, le temps est minimisé par le premier et maximisé par le deuxième ce qui implique des attitudes différentes des agents.

- **Langages de communication entre agents**

L'intérêt des langages d'interaction entre agents est de réduire les communications, parmi ces langages nous citons :

- **KIF** (Knowledge Interchange Format)

Dans le langage KIF [16] la syntaxe est inspirée de Lisp. Un message KIF peut être vu comme une s-expression, (soit un atome, soit une liste de s-expressions entourée de parenthèses).

- **KQML** (Knowledge Query and Manipulation Language)

Est un langage [16] de communication et un protocole de haut niveau pour l'échange de l'information, orienté messages et indépendant du mécanisme de transport (TCP/IP, SMTP, IIOP ou autre), aussi indépendant des langages KIF, SQL, Prolog ou autre.

- **FIPA-ACL** (Foundation for Intelligent Physical Agents):

Il se compose [16] d'un ensemble de types de message et des spécifications décrivent chaque acte communicatif avec une forme narrative et une sémantique formelle basée sur la logique modale.

8. Conclusion

Dans cette partie nous avons essayé de définir les concepts d'agents, agents intelligents, et leurs caractéristiques, outre de ces définitions nous avons défini les SMA et comment les agents communiquent entre eux.

B- Les agents mobiles

1. INTRODUCTION

Les agents mobiles sont aujourd'hui proposés comme un mécanisme intéressant pour implanter des applications sur Internet telles que la recherche d'informations ou le commerce électronique. Cette partie a pour objectif de donner une vision générale du paradigme agent mobile, en le comparant avec le paradigme client-serveur, nous présentons ensuite les travaux touchant à la notion d'agents mobiles.

2. DEFINITION D'UN CODE MOBILE

Ensemble des programmes capables de se déplacer pendant leurs exécutions. Un code mobile est exécuté de manière séquentielle ou parallèle sur une très grande diversité d'hôtes [NISE 02].

3. DEFINITION D'UN AGENT MOBILE

Un agent mobile est bien un code mobile capable de se déplacer de manière autonome d'une machine à une autre ou d'un site à un autre, il transfère son code, ses données, et son état d'exécution pour accéder à des données ou à des ressources, ou bien pour faire exécuter des tâches. Il est réactif, proactif et peut s'adapter à son environnement. [2]

4. LE CYCLE DE VIE D'UN AGENT MOBILE

Le cycle de vie d'un agent mobile se caractérise par sa création dans un contexte bien donné. Puis, lui attribuer un agent *proxy*¹ à des fins de communication. Ensuite, cet agent est préparé et expédié à une destination qui est le lieu de l'exécution de la tâche assignée. Ainsi dans le contexte du lieu d'arrivée, dans les limites de sécurité imposées, l'agent **rempli sa tâche** puis avise (retourner à la case départ ou poursuivre ailleurs). Néanmoins, certains événements peuvent interrompre momentanément l'agent et déclencher une désactivation quitte à entamer une

¹ Un proxy dans un système d'agents mobiles est un *agent intermédiaire* qui sert à assurer la transparence de la mobilité, faciliter à la fois le dépistage (localisation), réaliser la communication inter-agent et garantir la protection de certains éléments du code de l'agent.

réactivation quand certaines conditions seront remplies plus tard. Quand l'agent mobile a accompli la tâche qui lui a été déléguée, il retourne à la case départ pour rendre compte des résultats à celui qui l'a mandaté. Finalement, l'agent est soit retiré du contexte courant, soit détruit définitivement.

5. CARACTÉRISTIQUE D'UN AGENT MOBILE

5.1. La mobilité

Est la migration d'un programme avec restauration de son état initial, elle sert à augmenter la fiabilité et le bon fonctionnement des systèmes distribués.

5.2. Réduction de la charge dans le réseau

Les systèmes distribués [DANN 98] dans le paradigme client/serveur génèrent de nombreux messages pour exécuter leurs tâches (Figure I.6), mais avec des agents mobiles on les envoie sur le site où les tâches se font, les messages échangés deviennent locaux et libèrent d'autant la charge de réseau (Figure I.7) ce qui permet :

- ✓ de s'affranchir le temps de réponse.
- ✓ la réduction des délais de communication.

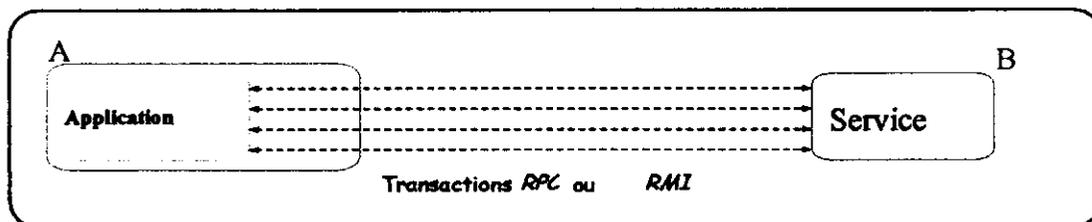


Figure I.5 Le paradigme client/serveur [DANN 98].

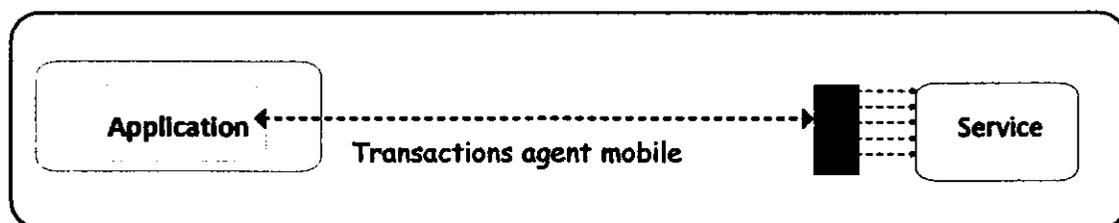


Figure I.6 : Le paradigme des agents mobiles [DANN 98].

5.3. Exécution asynchrone et autonome :

Le fonctionnement de paradigme client/serveur est *synchrone* du type *Requête-Réponse*, lorsque une machine cliente établit une connexion avec une machine serveur, lui envoie une requête et se met en attente de la réponse (Figure I.8), contrairement avec le paradigme agent mobile, un agent peut être envoyé sur le terminal mobile au moment de la connexion et travailler sur ce dernier après avoir interrompu la connexion. Le même agent peut attendre la prochaine connexion pour envoyer des informations sur le réseau signifiant par exemple qu'une tâche a été terminée (Figure I.9).

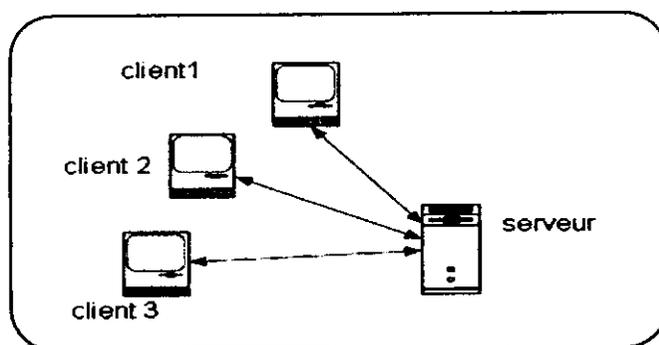


Figure I.7 Architecture Client/Serveur. [NICE 02]

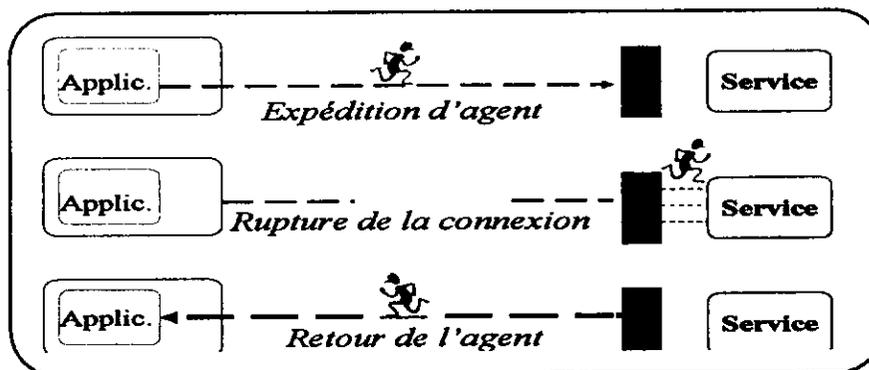


Figure I.8 L'asynchronisme des agents mobiles [DANN 98].

5.4. Adaptation dynamique à l'environnement

Les agents mobiles savent détecter les changements dans leur environnement, y réagir de manière autonome et s'adapter en conséquence. Ils ont la propriété de se répartir uniformément entre les machines d'un réseau pour résoudre un problème complexe de manière optimale.

5.5. Robustesse et tolérance aux fautes

La propriété des agents mobiles à réagir de manière dynamique aux situations et événements peu favorables, permet la conception des systèmes répartis robustes et tolérants aux fautes. Lorsqu'une machine hôte est en difficulté (rupture de communication, crash), les agents mobiles visiteurs prévenus ont la possibilité de se dispatcher ailleurs dans le réseau. Par contre dans le modèle *client/serveur*, si le serveur tombe en panne, la session de travail collaboratif ne peut plus continuer.

5.6. Communication

Dans le modèle client/serveur, lorsqu'un message passe d'un client à un autre, deux transmissions sont ainsi nécessaires, du premier client au serveur, puis du serveur au second client [EFBF 04]. Mais les systèmes basés sur les agents mobiles, leurs agents peuvent interagir avec d'autres agents pour atteindre leurs buts locaux ou globaux. Donc la communication permet la coopération et la coordination entre eux.

Tableau I.2 comparaison entre les modèles client/serveur et agents mobile.

	Client/serveur	Agents mobiles
Communication	Transmission indirecte des messages, ils passent tous par le serveur. Deux clients ne peuvent pas dialoguer entre eux directement [NIES 02]	Transmission directe des messages.
Gestion du groupe	Gestion centralisée du groupe [NIES 02].	La gestion des groupes peut être centralisée ou distribué
Tolérance aux pannes	La session [NIES 02] de travail collaboratif se termine si le serveur tombe en panne.	La panne d'une machine ne pénalise pas la session collaborative
Sécurité	Les machines des participants ne sont que des clientes [NIES 02].	Les agents offrent plus de sécurité
Gestion des messages	Le serveur peut faciliter la gestion des messages [NIES 02].	Gestion distribuée des messages

6. LA MIGRATION

La migration [DTHA 02] permet le transfert d'un agent d'un site à un autre à travers le réseau.

Il existe deux types de migrations dans les plates formes agents existantes:

6.1. La migration forte

Permet à un agent de se déplacer quelque soit l'état d'exécution dans lequel il se trouve et de reprendre son exécution après la migration exactement là où elle en était avant.

6.2. La migration faible

Fait transférer avec l'agent son code et ses données mais avec perte des traitements en cours.

7. Fonctionnement

Un agent part d'un [FRAN 02] site initiateur à travers un support de communication comme un message puis visite un ensemble d'hôtes pour y effectuer un traitement donné. Finalement l'agent revient sur le site de départ pour rapporter ces résultats.

8. LA SECURITE DANS LES SYSTEMES D'AGENTS MOBILES

Un agent mobile peut être cible à plusieurs types d'attaques du fait de son déplacement à travers des milliers de sites qui peuvent ne pas être de confiance et à travers un réseau qui n'est pas toujours sécurisé. Au cours de son déplacement, l'agent interagit avec d'autres agents qui peuvent analyser ou altérer son contenu. Aussi le site visité et sur lequel l'agent s'exécute a la possibilité de manipuler l'agent ou même de le détruire. Les aspects de sécurité concernant un agent mobile peuvent être résumés dans les points suivants [HOFR 97, 18]:

- **La sécurité de l'agent visiteur** : Il s'agit de protéger l'agent contre les manipulations d'un hôte malicieux.
- **La sécurité de l'hôte visité** : il est essentiel d'assurer la sécurité des données d'une machine hôte contre un programme visiteur étranger dont ni les intentions réelles, ni la crédibilité ne sont encore établies.
- **L'infrastructure de communication** : le transfert des données précieuses par un agent dans une infrastructure de communication hétérogène requiert un minimum de garantie contre les attaques, modifications et les ruptures

brutales de liaison. Dans ce cas il s'agit de la sécurité des agents pendant leur déplacement.

- **Sécurité entre deux agents** : Il s'agit de sécuriser un agent contre les attaques qu'il peut subir de la part d'un autre agent. On distingue **les attaques passives** qui ne modifient pas le code de l'agent ainsi que les données qu'il transporte tel que, l'espionnage des données transportées ; et **les attaques actives** qui consistent à l'altération du code ou des données de l'agent, par exemple la modification des valeurs des variables, ou toute modification du code. (Nous reviendrons en détailles à la sécurité dans le chapitre III).

9. QUELQUE SYSTEMES D'AGENTS MOBILES EXISTANTS

9.1. Le système Telescript

Est un système d'agents mobiles [JAME 95] pour le commerce électronique, proche de langage LISP intégrant la notion d'objets. Il propose un environnement utilisant différents concepts (Figure I.11). Les places sont constituées par les zones de services et les stations clientes du réseau.

Les agents sont des entités capables de migrer pendant leur exécution de place en place par l'instruction `go`, de rencontrer d'autres agents sur la même place par l'instruction `meet`, et de communiquer avec d'autres agents sur d'autres places par des connections permettant l'envoi de messages.

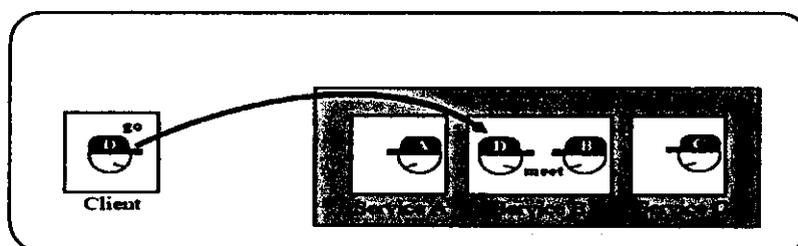


Figure I.9: Le système Telescript [JAME 95].

9.2. Le système Aglets²

Propose [LANG 96, DLMO 98] un environnement de programmation (plateforme) d'agents mobiles dans le langage Java y compris un ensemble d'abstractions et de comportements pour le développement sur Internet. Ainsi la migration est réalisée sans transfert implicite de l'état d'exécution (code et données seuls).

Le contexte³ de système aglets fournit des mécanismes de sécurité par l'intermédiaire d'un gérant reposent sur trois niveaux : la sécurité de Java, la sécurité des Aglets avec la notion de contexte, et la sécurité applicative grâce aux mécanismes d'événement inclut dans les Aglets.

9.3. Agent-Tcl

Est un système d'agents [GRAY 96] mobiles composé de deux programmes: un serveur qui s'exécute sur chaque site du réseau et un interpréteur.

✓ Le serveur reçoit puis authentifie et démarre les agents se présentant sur le site avec L'interpréteur, stocke les messages, fournit un service de désignation local et maintient une table d'information sur les agents présents sur le site.

✓ L'interpréteur fournit un mécanisme de sauvegarde et de restauration de l'état d'une exécution d'un script Tcl. Il propose des commandes pour la migration, la communication et la création d'agent.

Le cycle de vie d'un agent AgentTCL est illustré par la figure suivante :

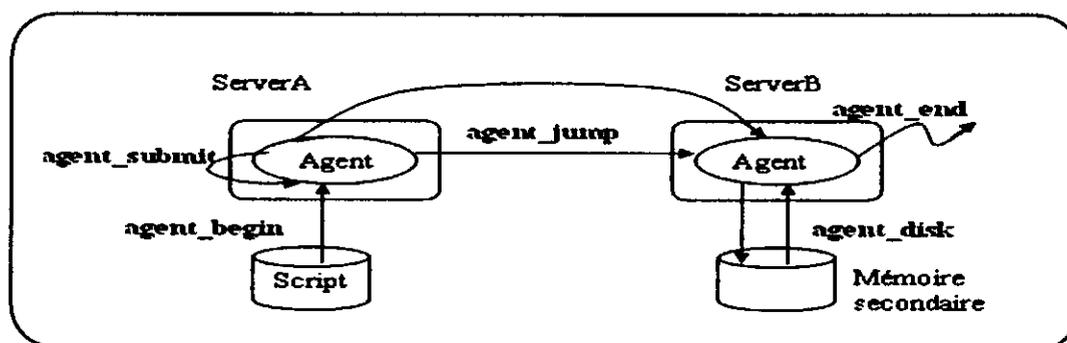


Figure I.10 Le cycle de vie d'un agent Agent-Tcl [GRAY 96].

²Un *aglet* est un objet *Java* comportant un flot d'exécution et disposant de méthodes pour le déplacement, le clonage, le rapatriement, la désactivation sur support persistant, l'activation depuis un support persistant et la destruction.

³ Un *context* est un espace d'adressage sur une machine pouvant abriter une ensemble d'aglets, il fournit un environnement sécurisé pour l'exécution des aglets, il peut y avoir plusieurs contextes par machine.

- ✓ La commande **agent_begin** permet d'enregistrer l'agent auprès du serveur, et lui associer un identificateur unique.
- ✓ La commande **agent_submit** permet d'instancier un agent fils avec un script Tcl sur une machine du système, puis le système envoie le script de l'agent au serveur de la machine cible qui démarre un nouvel agent avec l'interpréteur.
 - ✓ La commande **agent_jump** permet de migrer l'agent sur une autre machine du réseau. L'agent migré, est crypté et signé digitalement.
 - ✓ Les commandes **agent_send** et **agent_receive** permettent la communication asynchrone avec des messages.
 - ✓ Les commandes **agent_meet** et **agent_accept** permettent l'établissement de rendez-vous synchrone entre deux agents, puis réaliser des communications directes sur une connexion identifiée (TCP) plutôt que d'attendre des messages dans un buffer maintenu par le serveur.

9.4. TACOMA

Il Fournit [JVRS 95] un support dans les systèmes d'exploitation pour la programmation d'agents mobiles. Il ne fait pas de distinction entre un client, un serveur et un agent. C'est au niveau applicatif que l'on peut instancier un agent avec un comportement de client mobile, de serveur statique ou réaliser un monde d'agents intelligents.

Un agent est un processus exécutant un script Tcl capable de se déplacer volontairement sur une autre machine, ainsi peut manipuler les données locales et transporter des données lors de ses déplacements. TACOMA introduit pour cela la notion de **Folder** qui correspond à une unité de donnée manipulable par un agent.

Chaque Folder est identifié par un nom et contient une valeur. Localement un agent manipule des Folders par l'intermédiaire d'un **File Cabinet** qui est un espace de stockage persistant. Lorsqu'il se déplace, il transporte des Folders par l'intermédiaire d'un **Briefcase** mobile. La figure ci-dessous illustre ces concepts :

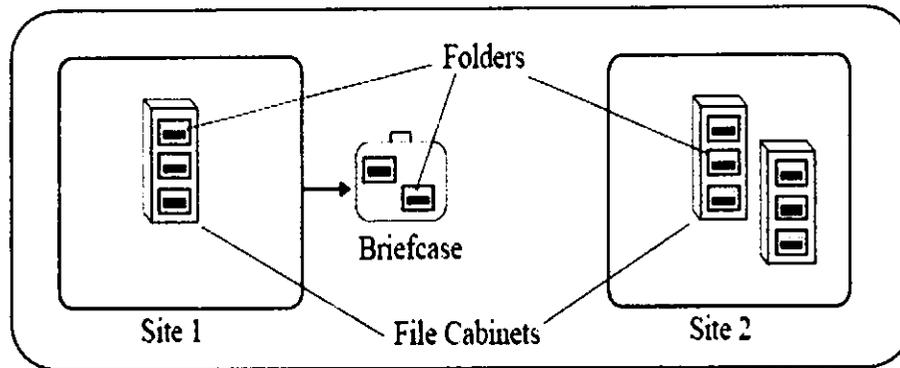


Figure I.11: Les Folders de TACOMA [STEP 97].

La communication avec un agent distant (Figure I.16) est fondée sur le concept du rendez-vous par la commande `meet` à l'intermédiaire d'un agent mobile nommé `rexec`, en utilisant `Briefcase` contenant des `Folders HOST` pour désigner le site cible, et `CONTACT` pour spécifier l'agent distant.

Sur le site distant un agent système nommé `tac_firewall` réceptionne le `Briefcase` qu'il conserve sur disque et instancie un agent système nommé `tac_exec` qui va jouer le rôle de représentant pour le rendez-vous distant [STEP 97].

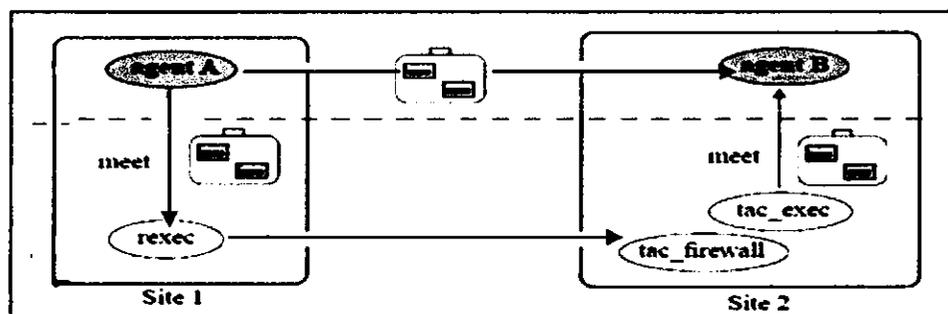


Figure I.12: Rendez-vous distant de TACOMA [STEP 97].

9.5. CONCORDIA

Est [CONC 00] un environnement de développement d'exécution et de gestion d'agents mobiles. Il est composé d'un ensemble de classes Java permettant l'exécution de serveurs, le développement d'agents et leur activation.

L'architecture d'un serveur Concordia est composée de :

- ✓ **Security Manager:** c'est le responsable de la sécurité du système Concordia.
- ✓ **Agent Manager:** Fournit le support de base pour la mobilité des agents, un support d'administration, un contexte d'exécution ainsi que des moyens à l'agent pour qu'il puisse contrôler son propre itinéraire.
- ✓ **Inter-Agent Communications Manager :** d'une part, ce composant veille à la collaboration des agents pour qu'ils puissent se partager les données d'une manière synchrone et cohérente; d'autre part, il assure l'envoi des événements distribués (interactions, modifications de l'état des agents).
- ✓ **Administrator:** il est responsable du lancement ou de l'arrêt des composants du serveur Concordia. Il permet également de gérer les modifications concernant les profils de sécurité des agents et des serveurs tout en permettant l'envoi de requêtes au Security Manager pour le compte d'un agent ou d'un serveur.
- ✓ **Service Bridge:** Permet au développeur d'applications agents d'ajouter des services à un serveur Concordia.
- ✓ **Directory Manager :** les agents ont recours à ce composant pour localiser les serveurs avec lesquels ils souhaitent interagir. Les serveurs d'applications exportent leurs services à destination des agents mobiles en s'enregistrant auprès du Directory Manager.
- ✓ **Queue Manager :** ce composant est responsable de la mise en place des files d'attente d'agents entre Agents Managers résidents sur des réseaux différents.
- ✓ **Persistence Manager :** Permet de sauvegarder l'état des agents et des serveurs sur disque.

9.6. Voyager

Est [OBSP 97] un ORB (Object Request Broker) doté d'un SMA, il permet aux programmeurs Java de créer rapidement et facilement des applications réseau sophistiquées en utilisant à la fois les techniques traditionnelle, et les techniques avancés d'objets distribués grâce aux agents.

10. JAVA : LANGAGE DE PROGRAMMATION DES AGENTS MOBILES

La notion d'agent mobile pose un problème d'intégration dans les langages. Le système doit gérer l'isolement de l'agent de manière à ce qu'il devienne autonome pour sa mission. Les systèmes utilisant Java sont confrontés d'une part au problème d'isolation du code et des données de l'agent et d'autre part au problème de la migration des threads. Ils résolvent le premier par la sérialisation : l'agent emmène avec lui une copie de tous les objets qu'il référence. Et ils contournent le deuxième en redémarrant l'agent avec un seul flot d'exécution sur une procédure d'entrée unique. C'est à dire que l'état d'exécution n'est pas complètement restauré [21].

11. AVANTAGES ET INCONVENIENTS DES AGENTS MOBILES

➤ La performance

Les premières améliorations apportées par les agents mobiles portent sur le gain de performances dues à une meilleure utilisation des ressources physiques mises à disposition, dont nous distinguons :

○ La diminution de l'utilisation du réseau

Le déplacement des agents mobiles permet de réduire les communications distantes entre les clients et les serveurs. En privilégiant les interactions locales, l'utilisation du réseau va se limiter principalement au transfert des agents. Cette situation permet :

✓ La diminution des temps de latence [DJOH98, JXHX 02, KRSG99]. Dans le contexte des réseaux à large échelle, la mise en place d'applications réparties, nécessitant de fréquentes interactions entre client et serveur, il arrive fréquemment que le temps d'attente de la réponse d'une requête soit plus long que le temps de traitement nécessaire à la réalisation du service. En rapprochant client et serveur dans un même sous-réseau, voire sur un même site, on les place dans un environnement où les temps de réponse des interactions sont limités, ce qui permet de réduire d'autant les temps de latence.

✓ De réduire le plus possible les communications distantes aux seuls transferts d'agents mobiles, on diminue considérablement les périodes de connexion entre deux sites.

○ **Des calculs indépendants**

Dans le modèle classique d'évaluation à distance [GCKR 01], le client et le serveur doivent rester connectés tant que le service est en cours d'exécution. En plus des problèmes liés aux fluctuations des performances réseaux, certains services, nécessitant de longues phases de traitement, ne supportent pas facilement une rupture de connexion avec le client. Dans ce cas, ils doivent souvent redémarrer entièrement leurs calculs. Mais, le maintien du lien de communication peut s'avérer difficile dans des réseaux à large échelle ou sans fil. Avec les agents mobiles, un client peut déléguer les interactions avec le service sans maintenir une connexion de bout en bout [GCKR 01]. Avec cette possibilité d'un calcul indépendant, le client peut demander un service, se déplacer (ou simplement terminer une session) puis venir récupérer les résultats plus tard.

○ **Optimisation du traitement**

L'optimisation des phases de traitement [LIDH 99] se produit à deux niveaux :

✓ Suppression des phases de dialogue entre le client et le serveur qui sont perturbées par des temps de latence dus aux communications réseaux.

✓ Le déplacement des agents va permettre de déléguer les calculs sur des machines serveurs, qui sont généralement plus puissantes qu'une machine cliente.

○ **Tolérance aux fautes physiques**

En se déplaçant avec leur code et données propres [SIRO 02], les agents mobiles peuvent s'adapter facilement aux erreurs systèmes. Si on prend le cas d'un site perdant une partie de ses fonctionnalités, un service tombant en panne, l'agent pourra alors choisir de se déplacer vers un autre site contenant la fonctionnalité désirée. Ceci permet une meilleure tolérance aux fautes que le modèle statique classique.

➤ La conception

Du point de vue de la conception [MDWH 99], avec la méthode classique, il va être très contraignant de décrire des algorithmes d'exploration (de réseau) ou bien encore de caractériser les déplacements des utilisateurs nomades.

Avec les agents mobiles, les concepteurs disposent d'une méthode qui permet de décrire naturellement ce genre de comportement. Ainsi, nous pouvons facilement mettre en place un déploiement et/ou une maintenance d'application sur un réseau ou encore suivre les utilisateurs dans leurs déplacements. Ensuite, les agents possèdent une capacité de traitement spécifique leur permettant de s'adapter à leur environnement. Généralement, dans le modèle client/serveur, le service est caractérisé préalablement par une interface stricte avec un protocole d'utilisation bien défini. Ainsi, si le client n'a pas connaissance de la description du service, il ne pourra pas l'utiliser. Par contre, les agents pourront eux s'adapter aux caractéristiques des services afin d'exprimer leur requête. Par exemple, si un service demande une communication sécurisée, l'agent pourra récupérer un module de sécurité, mettre à jour sa pile de communication et dialoguer avec le service. Nous pouvons aussi tout à fait imaginer que l'agent pourra s'adapter aux conditions du réseau en se séparant d'une partie de ses fonctionnalités pour s'adapter aux terminaux pauvres en ressources. La capacité de raisonnement va permettre de concevoir des agents qui seront autonomes, adaptant leurs déplacements en fonction de l'environnement et pouvant moduler leurs fonctionnalités en cours d'exécution. Cependant ces propriétés s'accompagnent d'une conception bien plus difficile que celle du classique client/serveur.

➤ Le développement

Le domaine des agents mobiles étant encore relativement jeune [DJOH 04], il se heurte à de fortes contraintes lors des phases de développement.

✓ La première est qu'il existe à l'heure actuelle un trop grand nombre de plateformes pour agents mobiles qui possèdent chacun leurs propres défauts et qualités, il est difficile de parler de standardisation et aucune ne semble encore s'imposer.

En sachant que les agents doivent s'adapter aux conditions de l'environnement, les développeurs sont confrontés à un éventail de possibilités bien trop large. Le manque de standardisation se retrouve aussi dans les interactions entre agents, ainsi il n'existe pas de langage partagé par toutes les plates-formes. Ceci représente un handicap sérieux, car les agents ont besoin d'exprimer les caractéristiques des services qu'ils recherchent et surtout d'obtenir des réponses précises sur leur localisations ainsi que sur la manière de les utiliser. De plus, avec tous les langages existants, les développeurs sont, encore une fois, face à un trop large éventail de possibilités.

✓ Le second problème important lors des phases de développement est la complexité de mise au point des programmes qui constitue une partie critique du processus de développement. La plupart des environnements de programmation possèdent des outils permettant de suivre les différents éléments d'une application durant son exécution afin d'en trouver les erreurs. Cependant, ce genre d'outil s'utilise parfaitement avec des éléments statiques mais est difficilement applicable dès le moment où les éléments se déplacent. Ainsi, il peut s'avérer impossible de récupérer l'état d'erreur d'un élément se trouvant sur un site déconnecté. Sans outil de débogage, il devient très difficile de différencier erreurs de conception et de développement.

✓ Le troisième problème important, en relation avec le débogage, est la difficulté de mettre en place une vérification des applications à base d'agents mobiles. La technique la plus utilisée encore de nos jours est le Test⁴. Cette méthode est déjà loin d'être simple dans le cas d'une application répartie statique, alors cela est encore plus compliqué si les points d'entrées (les agents) deviennent mobiles. Donc pour vérifier les applications construites sur des agents mobiles, il faut utiliser une méthode permettant de garantir le comportement et les interactions des agents avant leur déploiement. L'analyse statique⁵ de programme a fait ses preuves dans le cadre de la programmation classique mais elle constitue toujours un domaine de recherche pour les applications réparties.

⁴ Test : méthode sers à mettre l'épreuve d'une application en lui injectant des données dans ses différents points d'entrée dans le but de simuler le comportement utilisateur et de recouvrir une plage de situations la plus large possible.

⁵ Analyse statique : méthode permet de déterminer statiquement, sans exécuter un programme, des propriétés qui seront satisfaites lors de l'exécution en garantissant un certain niveau de correction.

➤ **La sécurité**

D'un point de vue logiciel [DRGC 91], la sécurité consiste à empêcher des accès, ou des modifications, non-autorisés aux éléments d'un système informatique. Lorsque nous souhaitons avoir un système à base d'agents sûr, nous devons mettre en place une politique de sécurité qui doit garantir la confidentialité⁶, et l'intégrité⁷.

12. Conclusion

Nous avons présenté dans cette partie l'approche d'agent mobile qui peut nous donner des résultats meilleurs que les autres technologies. Nous pouvons dire que le problème le plus important empêchant l'adoption actuelle des agents mobiles est sans nul doute la sécurité. En effet, même si la protection des sites est quasiment assurée, celle des agents qui n'a pas de solution définitive (voir chapitre III).

⁶ La confidentialité : les données des éléments ne sont pas divulguées aux demandeurs non autorisés.

⁷ L'intégrité : les éléments ne sont pas modifiés par des demandeurs non autorisés.

Chapitre II

L'e-commerce et les agents

A- Le commerce électronique

1. INTRODUCTION

Le commerce électronique aujourd'hui occupe une place de plus en plus prépondérante dans l'univers des télécommunications, car les activités telle que la collecte des données, la négociation, la prise de décision en groupe, l'établissement des plannings, la publicité, le financement des projets, le marketing, l'achat et le vente, etc....sont de plus en plus tributaires à l'Internet.

2. A QUOI SERT LE COMMERCE ELECTRONIQUE ?

Le commerce électronique influe la manière dont les utilisateurs finaux négocient, passent des commandes, paient pour des biens et services, et il soulève des questions au niveau juridique [22, LILL 06]. Du point de vue du consommateur, rien n'est plus simple qu'acheter depuis chez soi, n'importe quand et n'importe où dans le monde. Du point de vue du vendeur, les intérêts sont multiples. On peut citer la nécessité de e-commerce dans les points suivants : [PRES 99]

- ✓ La possibilité de comparer les offres facilement et efficacement.
- ✓ Le gain du temps et des frais.
- ✓ Plus pratique et plus agréable.
- ✓ On peut consommer quand on veut.
- ✓ suppression éventuelle des intermédiaires entre producteur et consommateur.
- ✓ vaste clientèle potentielle non limitée géographiquement
- ✓ Il nous offre d'avantage de choix.
- ✓ Des horaires d'ouverture 24 h sur 24.
- ✓ Des produits livrés directement à domicile.

3. DEBUT DE COMMERCE EN LIGNE

Pendant les dernières années, le nombre de sites conçus pour vendre des marchandises sur Internet n'a pas cessé de croître. Au début, il s'agissait de sites web simples qui montraient quels produits étaient disponibles, sans vraiment supporter de transactions en ligne, et les communications avec les clients étaient faites à travers les E-mails. Le problème principal avec ces E-mails publicitaires est qu'ils n'étaient pas ciblés, ainsi la plupart du temps, les personnes les recevant ne se sont pas

intéressées. Pire, elles se sentent dérangées par ces méthodes brutales et protestent, à juste titre [33].

4. DEFINITIONS DE L'E-COMMERCE

Il n'y a pas de définition universelle du commerce électronique en raison du grand nombre de marchés et d'acteurs sur Internet et de l'évolution rapide de leurs rapports complexes. [23] citons les définitions suivantes :

- **Définition 1**

On appelle « **Commerce électronique** » (ou E-commerce) l'utilisation d'un média électronique pour la réalisation de transactions commerciales. La plupart du temps il s'agit de la vente des produits à travers le réseau Internet. [24]

- **Définition 2**

L'E-commerce désigne l'ensemble des échanges commerciaux dans lesquels l'achat s'effectue sur un réseau de télécommunication. [25]

- **Définition 3**

L'E-commerce permet sous une forme électronique et grâce à une communication à distance interactive, la conclusion en ligne d'un contrat de vente ou de prestation de services, et leur livraison (ou téléchargement) après le paiement du prix convenu [MICH 98].

- **Définition 4**

L'E-commerce est avant tout du commerce et Internet n'est qu'un moyen ou support de communication, comme entre autres, l'EDI (échanges des données informatisées), le Minitel, etc. Il recouvre toute opération de vente de biens et de services via un canal électronique. [25]

5. TYPOLOGIES DU COMMERCE ELECTRONIQUE

Les applications de e-Commerce tombent dans une des deux principales catégories bien connues :

5.1. Business to Business (B2B) : où le client est une autre organisation, autrement dit il s'agit du commerce inter entreprise, et la nature des relations est caractérisée par une stabilité à long terme [PRES 99].

5.2. Business to Customer (B2C) : l'entreprise vend ses produits et services aux consommateurs finaux via un site intermédiaire. [25]

A côté des deux grands ensembles commerciaux cités ci-dessus, on peut identifier d'autres types de relations moins important car ne générant pas des revenus élevés:

5.3. Consumer to Business (CtoB) : Son principe est de regrouper les consommateurs pour acheter en gros à moindre prix, via un site d'achat [25].

5.4. Consumer to Consumer (CtoC) : s'applique [25] aux échanges commerciaux entre des personnes privées dans lesquels un site spécialisé joue le rôle d'intermédiaire.

5.5. Business to Government (BtoG) : relation concernant les transactions électroniques entre une entreprise et une administration gouvernementale, par exemple la transmission d'une déclaration fiscale [25].

5.6. Government to Consumer (GtoC) : relation touchant les transactions électroniques entre une personne privée et une administration gouvernementale, par exemple la mise en ligne de formulaires administratifs ou la possibilité de payer ses impôts par Internet [25].

5.7. Employee to Employee (EtoE) : relation touchant les échanges électroniques entre au moins deux employés d'une même organisation ou d'organisations différentes au travers d'un Intranet ou d'Internet [25].

6. LES DIFFERENTES TACHES DE L' E-COMMERCE

Le commerce électronique [LILL 06] peut être considéré comme un large domaine décomposable en un grand nombre de tâches plus ou moins atomiques. Les trois phases principales dans le processus d'interaction entre un client et une organisation dans le cadre du e-Commerce :

- ✓ L'avant vente.
- ✓ La vente.
- ✓ L'après vente.

Le Tableau II.1 montre les activités utilisateurs qui pourront être effectuées dans les trois phases:

Tableau II.1 Les différentes phases en e-commerce [LILL 06]

Phases	Activités Utilisateurs	Accès
Avant vente	<ul style="list-style-type: none"> - Voir promotion - Accepter promotion - Rechercher produit - Comparer Produit - Evaluer Produit - etc. 	<ul style="list-style-type: none"> -Télévision - Appel téléphonique - e-Catalogue - Forum - Formulaire électronique - etc.
La vente	<ul style="list-style-type: none"> - Accepter vente - Commander produits - Négocier le prix - Choisir et informer le moyen de paiement - Choisir et informer le mode de livraison - Confirmer l'achat - Conserver une trace - Constituer un dossier - etc. 	<ul style="list-style-type: none"> - Appel téléphonique - Formulaire - Boîte de dialogue - Système d'authentification - Annuaire - e-Mail/SMS notification - etc.
Après vente	<ul style="list-style-type: none"> - Accepter la livraison - Renvoyer le produit - Mettre en œuvre le produit - Développer des usages - Réclamer - Faire dépanner - Revendre le produit - Acheter des compléments au produit, etc. 	Document riche média

7. COMPARAISON ENTRE LE COMMERCE ELECTRONIQUE ET LE COMMERCE TRADITIONNEL

Le cycle de vente d'une transaction traditionnelle est similaire à celui d'une transaction électronique, toutefois, les méthodes d'obtention et de transmission de l'information varient.

Dans une transaction traditionnelle, de multiples vecteurs de communication sont indispensables. Cette diversité a pour conséquence de compliquer la coordination des opérations et d'allonger considérablement le temps de traitement d'une commande. En revanche, dans le cas d'une transaction électronique, l'information est numérisée de bout en bout et il n'existe qu'un seul vecteur de communication : le Web.

Avant, l'acheteur faisait une demande mais il passe à la commande qu'après avoir obtenu l'accord, choisi le fournisseur chez qui le produit est disponible en se basant sur des formulaires imprimés ou des lettres ; tandis que le commerce électronique permet à l'acheteur de choisir le produit à partir d'un site Web, exploiter les techniques électroniques pour effectuer sa demande et la suivre, en utilisant un unique moyen de communication (**Web**) [33].

L'achat sur Internet présente de grandes différences avec un achat dans un supermarché ou une boutique. Il se distingue également d'une commande dans un catalogue.

8. L'ACHAT EN LIGNE

Certains sites commerciaux sont avant tout des vitrines d'information. D'autres permettent de placer des commandes tout en payant hors-ligne. Aujourd'hui, les sites sont plus évolués et permettent de payer en ligne; il s'agit d'une fonction utile pour les produits dont on désire la livraison immédiate. [27]

Il existe cependant quelques freins de l'achat en ligne [YAKL 02] tel que :

- ✓ La réticence à payer en ligne.
- ✓ La crainte de ne pas être conseillé.
- ✓ La crainte que la commande ne soit pas livrée à temps.
- ✓ Le besoin de toucher les produits.
- ✓ Le manque d'attractivité des prix.
- ✓ La crainte d'une violation de données personnelle.

9. PLACE DE MARCHÉ (MARKET PLACE)

Il s'agit d'un espace (plate-forme) de rencontre virtuel, sur Internet où vendeurs et acheteurs se rencontrent à fin d'arriver des ententes, signer des contrats, et effectuer des transactions. La place de marché est régie par des règles publiques définissent ce qui est permis et non sur elle, aussi, ce que les uns et les autres doivent ou ne doivent pas faire [CHSI 98]. Les places de marché facilitent les transactions, particulièrement pour les produits dont le prix varie beaucoup ou est incertain, ou les produits qui nécessitent plusieurs offres.

10. LES CARACTERISTIQUES DU COMMERCE ELECTRONIQUE

L'intérêt [MICH 98] croissant des entreprises pour le commerce électronique a obligé les grands concepteurs de logiciels à concevoir des outils pour faciliter la tâche des entreprises. Les magasins virtuels étaient constitués de pages HTML statiques qui référençaient les produits ce qui oblige le client de noter les références des produits qui l'intéressent avant de remplir un formulaire ou d'envoyer la commande par E-mail ; le paiement était fait en communiquant son numéro de carte de crédit bancaire.

Cette méthode comportait de nombreux inconvénients, le principal est le manque de sécurité dans la transaction. Pour pallier à tous ces problèmes, consiste à fournir à l'entreprise un package complet qui va lui permettre de gérer son serveur de commerce électronique. Ce type de solution permet de concevoir le magasin virtuel, de gérer les accès, de contrôler et d'assurer le paiement.

Une autre caractéristique est la possibilité d'utiliser une large gamme de bases de données. En effet, toutes les entreprises possèdent une base de données décrivant leurs stocks, leurs fournisseurs, leur client,...

11. LES PROBLÈMES DE SÉCURITÉ DES TRANSACTIONS SUR INTERNET

Certains obstacles ont freiné le développement du commerce électronique. Dans un monde où le contact n'est établi qu'à distance, les systèmes de paiement doivent offrir un haut niveau de sécurité et créer un climat de confiance.

Les problèmes liés à la sécurité des transactions sur Internet sont les suivants [28,29]:

11.1. La confidentialité

Les renseignements personnels transmis par Internet doivent demeurer confidentiels, et seul le destinataire puisse les lire. Dans ce cas l'utilisation d'un outil de *chiffrement*⁸ est un moyen efficace d'assurer la confidentialité des données

⁸ Le chiffrement est un outil simple qui consiste à transformer des mots et des nombres significatifs en langage codé

11.2. L'Authenticité

Pour éviter les fraudes, il nous faut vérifier l'identité de chacun des participants à une transaction sur Internet à l'aide de mots de passe, de *certificats numériques*⁹ de *signatures numériques*¹⁰ ou de secrets partagés (renseignements connus juste par les deux participants communs à une transaction).

11.3. Non-répudiation

Grâce aux *signatures numériques*, les participants ne peuvent pas faire une transaction sur Internet, puis nier ensuite l'avoir faite.

11.4. L'intégrité

L'information transmise ou accessible sur Internet ne peut pas être modifiée. Dans le cas d'un message envoyé par Internet, son contenu doit être le même au moment de la réception qu'au moment de l'envoi. Les *signatures numériques* peuvent servir à confirmer l'intégrité des données, car elles détectent tout changement dans celles-ci.

11.5. Contrôle de l'accès

Seul le personnel autorisé peut accéder aux données, pour ce là Les *pare-feu*, les droits d'accès, les mots de passe et les certificats d'autorisation sont des outils utilisés pour le contrôle de l'accès.

11.6. Disponibilité

L'information transmise sur Internet doit l'être au moyen de canaux sûrs et d'équipements de stockage fiables, qui sont opérationnels lorsqu'on en a besoin. La sauvegarde des données, la surveillance ainsi que l'utilisation de *logiciels antivirus* et de ressources informatiques adéquates sont des mesures de sécurité qui assurent la disponibilité des données et des services.

12. MODES DE PAIEMENTS

Un paiement [30] par Internet peut s'effectuer via différents modes, dans lesquelles nous trouvons:

⁹ Le certificat électronique (autorité) : est un document d'identité électronique attestant du lien entre une identité et une clef publique.

¹⁰ La signature numérique : est le cryptage d'un document électronique réalisé par une clef asymétrique privée.

12.4. Crédit

Le crédit est offert soit par *le commerçant* (le paiement par chèque dont le commerçant fait crédit jusqu'à présentation du chèque), soit par *la banque de l'acheteur* (les cartes de crédit et les chèques de voyage)

12.5. Porte-monnaie électronique

Le porte-monnaie est une mémoire portative, implantée sur une carte à puce, une carte PCMCIA, une disquette, ou tout ce qu'on pourra inventer. Cette mémoire, réputée inviolable, conserve le compte d'un argent fictif.

On remplit le porte-monnaie par une opération de retrait bancaire ou bien il se remplit tout seul par un crédit tournant. Le porte-monnaie électronique peut donc fonctionner, comme les cartes en mode crédit. L'argent électronique peut également passer d'un porte-monnaie à un autre.

13. MODES DE SÉCURISATION DE PAIEMENT

La sécurisation des paiements par carte bancaire sur Internet est l'un des problèmes majeur de la nouvelle économie. Afin d'apporter des solutions à tous ces problèmes de sécurité, authentification et confidentialité, des techniques ont été mises en place [30] :

13.1. SSL (Secure Socket Layer)

Ce protocole de communication a pour but de crypter les données qui transitent sur Internet et une authentification des postes clients et serveurs, il n'assure aucune gestion de la transaction; donc il nécessite d'avoir un terminal de paiement pour pouvoir encaisser les paiements par carte.

13.2. SET (Secure Electronic Transaction)

S'appuie sur le cryptage des données. Il offre cependant l'intérêt de protéger à la fois les données transmises par le client lors de la transaction et d'empêcher l'envoi des données bancaires confidentielles au logiciel marchand du commerçant. Les données bancaires sont directement envoyées à l'organisme bancaire chargé

d'encaisser les paiements et après vérification, le commerçant reçoit uniquement le détail des commandes qu'il doit traiter.

13.3. C-SET (Chip Secure Electronic Transaction)

Il s'appuie sur SET et gère la carte à puce des cartes bancaires. Dans ce protocole, l'utilisateur de la carte est identifié par un code secret.

14. DÉFINITION DE LA NEGOCIATION

La négociation est l'ensemble des démarches et des processus de communication ayant pour but de confronter les positions, à fin de parvenir à un accord entre les parties concernées. Autrement dit la négociation illustre l'art de commercer.

15. AVANTAGES DU COMMERCE ÉLECTRONIQUE

L'e-commerce ne se limite pas à la seule vente en ligne, mais peut servir également à [YAKL 02]:

- ✓ **Communiquer** : on peut utiliser le courrier et accéder à Internet.
- ✓ **Promouvoir** : les entreprises se servent de pages Web pour donner à des clients actuels et éventuels des renseignements sur la nature et la valeur de leurs activités. Les pages Web tiennent les clients au courant des produits, des services et des nouveautés des entreprises, et permettent à celles-ci de répondre aux questions de la clientèle. Les entreprises peuvent aussi utiliser leurs sites Web pour faire des études de marché en obtenant des renseignements auprès de clients ou de visiteurs qui se présentent sur leurs sites.
- ✓ **Vendre** : les ventes de produits peuvent avoir lieu sept jours sur sept, et 24 heures par jour, dans le monde entier.
- ✓ **Sensibiliser** : grâce au marketing et à la publicité en direct.
- ✓ **Assurer la prestation de services** à la clientèle comme le service après vente et la communication avec le client.
- ✓ **Economiser** : en éliminant une partie des frais associés aux transactions effectuées sur papier et au courrier.

- ✓ **Accroître** l'efficacité en profitant de la rapidité des communications.
- ✓ **Livrer concurrence** à l'échelle planétaire sans devoir ouvrir des bureaux dans d'autres pays.
- ✓ **Soutenir** la concurrence d'entreprises beaucoup plus grosses, parce que personne ne peut devenir, d'après votre présence sur Internet, de la taille ou de la jeunesse de votre entreprise.
- ✓ **Repérer de l'information** ou des renseignements sur les compétiteurs.

16. AVENIR DE COMMERCE ELECTRONIQUE

L'éclatement de la bulle financière des nouvelles technologies, selon de nombreux experts, sonnerait le glas du commerce électronique. Après des années d'euphorie, où l'on attendait tout des nouvelles possibilités du commerce électronique, celui-ci serait devenu "un mirage sans lendemain". Dans sa conférence inaugurale.

Les difficultés rencontrées par les entreprises fanion de la nouvelle économie amènent de l'eau au moulin des détracteurs du commerce électronique. "Mais qu'en est-il vraiment?"

17. CONCLUSION

Après avoir consacré le premier chapitre à la présentation des agents mobiles et à la mise en comparaison des agents mobiles avec les systèmes classiques (Client/serveur) ; nous avons étudié, dans cette partie, le domaine de commerce électronique.

En ce qui concerne le commerce électronique, nous nous sommes attachés à présenter l'E-commerce en lui faisant une petite comparaison avec le commerce traditionnel, ainsi nous avons étudié les différentes taches de processus commercial en ligne.

B- L'utilisation des agents dans le commerce électronique

1. INTRODUCTION

Dans le commerce électronique, les agents intelligents commencent à prendre une place importante en vue d'automatiser un certain nombre de tâches. Il s'agit de rendre la comparaison des prix, la fonction d'achat, la fonction de vente, la négociation de plus en plus automatisée. Cette tendance permettra à l'entreprise d'améliorer la connaissance qu'elle a de ses clients pour mieux adapter son offre aux besoins de marchés.

2. POURQUOI LE COMMERCE ELECTRONIQUE UTILISE LES AGENTS

La technologie Internet se développe et devient un canal principal pour la vente et l'achat des biens, ce qui est besoin d'un système qui cherche les produits aux quelles nous nous intéressons (une classification selon certains critères), et qui négocie avec les différents marchands d'une manière automatique. Pour cela le commerce électronique utilise une nouvelle activité pour faciliter les échanges entre le vendeur et l'acheteur, cette activité joue le rôle d'intermédiaire qui est capable d'aider le vendeur ou l'acheteur à faire des recherches, des commandes, et la livraison des produits ; aussi l'e-commerce nécessite un environnement qui assure la sécurité et la confiance. Un agent intelligent peut assister ces activités à cause de leurs caractéristiques (voir chapitre I). [DGJP 99]

3. LES BENEFICES D'UTILISATION DES AGENTS DANS L'E-COMMERCE

- **Bénéfices pour le consommateur** : gain de temps, analyse d'une offre commerciale plus étendue, transparence des marchés, soutien d'un agent qui connaît mieux en mieux ses goûts [YAKL 02].
- **Bénéfices pour le distributeur** : localisation plus facile des boutiques, augmentation des ventes pour celles qui parviennent à analyser la demande [YAKL 02].

4. TYPES D'AGENTS DANS LE COMMERCE ELECTRONIQUE

Nous distinguons deux types d'agents dans le commerce électronique :

4.1. Les agents acheteurs

Sont des agents intelligents contrôlés par les clients interviennent pour faciliter le processus d'achat. Ils peuvent nous renseigner [31]:

- ✓ Sur la disponibilité d'un produit en menant une recherche par marque ou par catégorie.
- ✓ Sur l'identification des distributeurs : localisation d'un distributeur précis.
- ✓ En traitant les informations collectées, par exemple grâce à des tableaux comparatifs des offres présentées sur divers critères (prix, services, avis d'autres consommateurs).
- ✓ En établissant une présélection automatique d'articles en fonction des préférences du consommateur (priorité au rapport qualité - prix, au service, aux avis des autres consommateurs,...).

4.2. Les agents vendeurs

Contrairement aux agents acheteurs, les agents vendeurs interviennent pour aider le vendeur dans ses tâches qui consistent à connaître mieux le client et ses besoins particuliers. Ainsi les agents vendeurs demandent au consommateur de décrire son profil qui permet d'obtenir une offre réellement personnalisée et adaptée aux besoins de chaque client. Un agent vendeur ayant un produit à commercialiser va traverser le réseau à la recherche des clients intéressés par ce produit. Lorsque l'agent vendeur rencontre un agent client intéressé par ce type de produits, une transaction est alors négociée entre les deux agents.

Les différentes fonctions assurées par les agents vendeurs sont [32] :

- ✓ Enregistrement du profil et des préférences de l'acheteur.
- ✓ Enregistrement des demandes successives de l'acheteur sur son profil.
- ✓ calculer des recommandations sur l'évolution de l'offre commerciale grâce à des statistiques sur la demande globale des consommateurs.

5. LA NEGOCIATION ENTRE AGENTS

La place de marché ou Market place est un lieu de rencontre entre agents négociateurs où les vendeurs proposent des produits aux acheteurs qui contrôlent si la description correspond à leurs buts. Le processus de négociation en général est comme suit [YAKL 02]:

- 1- Initialement, tous les utilisateurs doivent être inscrits dans la place de marché afin de créer leurs agents pour acheter ou pour vendre un produit dans le serveur.
- 2- Un agent créé communique avec l'agent médiateur centrale (manipule des communications des agents) pour entrer son profil, et les détails du job qu'il cherche à la base de donnée du marché électronique.
- 3- Après avoir eu les offres des agents acheteurs et ceux de vendeurs, le médiateur assigne ces agents dans un lieu de rencontre de négociation si leurs profils sont similaires où seuls les agents avec les préférences similaires peuvent négocier entre eux.
- 4- Après avoir négocié assez suffisamment pour prendre une décision finale, la négociation se termine : dans un cas idéal, l'agent acheteur et celui de vendeur ont une convention, et l'agent médiateur va être informé pour qu'il puisse mis à jour la base de données. Chaque agent contacte avec son créateur et l'informer avec qui il se met en accord, et l'envoie les termes sur lesquels on est d'accord.

6. Le comportement d'achat

Il y a différentes étapes du processus commercial représentant le comportement d'achat du consommateur. Parmi eux citons six étapes fondamentales [NICO 66] :

6.1. Identification des besoins

Cette étape caractérise la détermination des besoins commerciaux ou de consommation non satisfaits ou non rencontrés en consultant le produit.

6.2. Courtage de produit (product brokering)

Concerne la recherche de l'information sur qu'il faut acheter pour satisfaire les besoins du consommateur. Cela englobe l'évaluation d'alternatives de produit en se basant sur des critères fournis par ce dernier. Le résultat de cette étape est un *ensemble de considérations* de produit.

6.3. Courtage de marchands (merchant brokering)

Cette étape combine l'ensemble de considération de l'étape précédente avec évaluation de l'information concernant les marchands, afin de déterminer à qui les produits nécessaires seront achetés. Cela inclut l'évaluation d'alternatives marchands basées sur des critères fournis par les consommateurs (prix, garantie, disponibilité, délai de livraison,...).

6.4. Négociation

Cette étape applique un protocole de négociation afin de fixer les conditions d'achat ou de vente des produits. Dans les marchés de détail, le prix et d'autres aspects de la transaction sont souvent fixés. Et dans d'autres (les marchés d'automobiles, les marchés locaux,...), la négociation de prix ou d'autres aspects de la transaction sont intégrés au courtage du produit et du marchand.

6.5. Achat et livraison

Cette étape signale la terminaison de l'étape de négociation. Dans quelques cas, le paiement et les options de livraison peuvent être influencés sur le produit et par le marchand.

6.6. Service après vente et évaluation

Cette étape détermine le niveau perçu de la qualité du service des produits achetés.

On utilisera les agents intelligents dans ces différentes phases à fin d'éliminer un grand nombre d'activités qui prennent beaucoup de temps, ce qui permettra de réduire les frais de chaque transaction.

7. APPLICATIONS RECENTES D'AGENTS INTELLIGENTS EN COMMERCE ELECTRONIQUE

Des projets de développement ont été réalisés au profit de la technologie des agents visant spécifiquement à promouvoir le commerce électronique.

7.1. Les agents du courtage de produits

Ce sont des agents contrôlés par les clients pour faciliter leur processus d'achat, en sélectionnant une liste de produits correspondants à une requête. Nous citons par exemple :

- **PesonaLogic** : est un outil qui aide les consommateurs à déterminer quoi acheter en les guidant dans un vaste espace de caractéristiques de produits. L'agent permet aux consommateurs de spécifier les contraintes applicables aux caractéristiques du produit recherché. Un moteur de recherche conçu pour satisfaire ces contraintes affiche ensuite une liste ordonnée des produits qui satisfont toutes les préférences du consommateur [PERS 01].

- **Firefly** : est un agent qui aide les consommateurs à trouver des produits. Mais au lieu de baser le filtrage sur les caractéristiques d'un produit. Le système est utilisé pour recommander des produits de consommation [FIRE 00].

7.2. Les agents du courtage des marchands

Ce sont des agents contrôlés par les clients qui comparent les prix offerts par plusieurs vendeurs pour un produit donné à fin de trouver l'offre la plus avantageuse. Ces agents de magasinage ne comparent pas des produits, mais des marchands.

- **Bargain Finder** : développé [KRUL 95] par Andersen Consulting, quand un utilisateur lui fournit le nom d'un produit particulier, Bargain Finder peut interroger plusieurs sites Web de marchands pour déterminer et comparer les diverses différences de prix.

- **Jango** : est [DOEW 97] un agent de négociation plus récent. Il a été développé pour corriger certaines limites des premières versions d'agents de courtage de marchands. Cet agent dépasse les blocages que rencontre Bargain Finder.

Au lieu de lancer les requêtes d'un site central comme Bargain Finder, Jango utilise les moteurs de recherche du client et les requêtes apparaissent comme celles d'un simple utilisateur [JANG 01, RICH 98] .

7.3. Les agents de négociation

Ce sont des agents capables de négocier des prix, des conditions de vente pour un produit déterminé par le client et avec une stratégie de négociation déterminée par le client.

- **Auction Bot** : de l'université du Michigan permet aux acheteurs et aux vendeurs de se réunir dans le même espace virtuel et de participer à des enchères personnalisées, créées par les vendeurs qui sont autorisés à spécifier des paramètres tels que les délais de soumission, les méthodes de résolution en cas d'égalité des offres, etc. L'une des caractéristiques censées distinguer Auction Bot de plusieurs autres sites de ventes aux enchères est son *interface programmable*, qui permet aux utilisateurs de créer leurs propres agents logiciels capables de participer de façon autonome aux enchères Auction Bot [RICH 98].

- **Tete-a-Tete(T@T)** : est l'une des innovations récentes de la technologie des agents pour le commerce électronique. La caractéristique qui distingue cette technologie de ses prédécesseurs est que T@T négocie de façon coopérative et non compétitive. T@T peut aussi négocier les diverses conditions d'une transaction, y compris les garanties, les délais de livraison, les contrats de service, les services de cadeau et autres services. Il enregistre les caractéristiques des produits pour faire un filtrage de l'offre sur Internet. [RGPM 98]

8. QUELQUES ARCHITECTURES DES SYSTEMES DE COMMERCE ELECTRONIQUE EXISTANTES A BASE D'AGENTS MOBILES

8.1. L'infrastructure Kasbah

➤ Définition

Kasbah est un site web du MIT Media Lab, où l'utilisateur achète et vend des biens, ainsi est un système multiagents proactifs dont les agents sont créés par l'utilisateur qui spécifie une certaines contraintes [ACPM 98].

➤ Fonctionnement de Kasbah

-**Agent vendeur** : quand un utilisateur crée un agent vendeur, il lui donne une description de l'article qu'il veut vendre, des paramètres pour le guider comme le prix désiré, le plus bas prix acceptable, et la date à laquelle la transaction doit être effectuée ; ces paramètres peuvent changés à n'importe quel moment après la création de l'agent, ils définissent le but de l'agent qui est la vente de l'article en question pour le prix le plus élevé que possible. Idéalement, c'est le prix désiré, mais le prix le plus bas accepté, c'est ce qui attire l'acheteur intéressé dans le cadre de temps donné [ACPM 98].

L'agent vendeur est autonome en lui-même. Une fois qu'il se trouve dans l'espace marchand, il négocie et prend des décisions seul sans l'intervention de l'utilisateur.

-**Agent acheteur** : quand un agent acheteur est créé, l'utilisateur décrit l'article qui l'intéresse en mettant des paramètres par exemple la date d'achat, le prix qu'il veut payer, et le plus haut prix qu'il peut accepter [ACPM 98].

-**La négociation:** est [35] basée sur un protocole demande-réponse et se fait strictement entre agents. Une fois les agents acheteurs et les agents vendeurs sont appariés, les agents acheteurs peuvent proposer une offre aux agents vendeurs, sans restriction de délai ou de prix, et ces derniers agents répondent par « oui » ou par « non ». Ainsi quand un agent exécute une transaction, un avis est envoyé à l'utilisateur qui a créé l'agent, lorsqu'un agent a conclu une affaire, il cesse de négocier avec les autres agents et demande automatiquement au marché de le retirer de la liste des agents actifs (i.e. les autres agents ne pouvant pas lui envoyer des messages).

Les méthodes suivantes permettent la communication directe entre les agents :

- ✓ **What-is-item?** : demander à un agent de spécifier l'article qu'il veut vendre ou acheter.
- ✓ **What-is-price ?** demander à un agent le prix de l'article. si l'agent vendeur qui demande le prix alors c'est le prix pour lequel il est prêt à vendre ; sinon la somme à payer.
- ✓ **Accept-offer ?** retourne accepte ou refuse

8.2. MAGMA (Minnesota Agent Market Place Architecture)

Est [TGMW 97] un système du place marché virtuel, il est créé pour faire transféré des marchandises sur internet. Il est basé sur plusieurs agents commerçants, le serveur de publicité, serveur de communication, et la banque.

- **Agents commerçants** : ils mènent leurs affaires dans ce système, ils sont responsables pour acheter et vendre des marchandises. L'agent commerçant est composé de :
 - **Le portefeuille (The wallet)** : facilite le paiement pendant l'opération d'achat, il contient tous les comptes bancaire, les chèques, les bons d'achat, ..., et les opérations a exécute sur eux.
 - **L'inventaire (The inventory)** : contient tous les informations des marchandises et leurs adresse URL, ces informations sont sauvegardées dans des bases de données ; ainsi les marchandises peuvent être ajoutés manuellement dans la base de donnée.

- **Ad-Manager** : permet à l'utilisateur de mettre des annonces pour vendre des marchandises de l'inventaire ; comme il autorise l'utilisateur d'enlever des annonces.
- **Le négociateur (The negociator)** : facilite la recherche des marchandises. La négociation soit s'effectue par la recherche automatique de produit, ou par agent manuellement commandés.

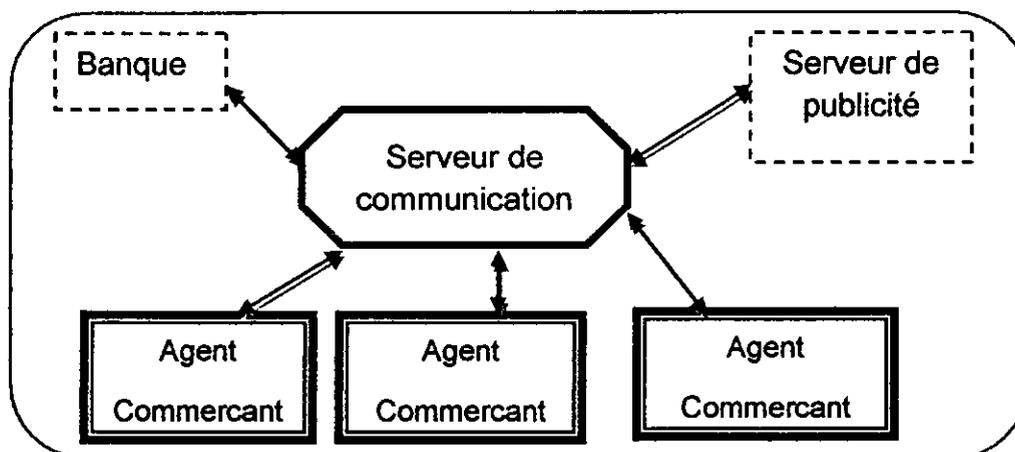


Figure II.1 L'architecture de MAGMA [TGMW 97].

- **Le serveur de publicité (Advertising server)**: fournit un service qui inclut la recherche d'annonce par catégorie, en utilisant des bases de données indexées avec un moteur de recherche ; cette base de donnée contient des mots clés, une description courte du produit, et les liens URL avec des informations supplémentaires d'acheteur.

- **La banque** : Le système bancaire est implémenté par le système d'agent électronique développé par Bigicash. Elle fournit des services comme les comptes, les chèques, les lignes de crédit..., qui sont sécurisés par la signature numérique de l'agent. De plus la banque garde les traces de contrôle qui contiennent tous les informations de transaction produisent dans une période de temps.

8.3. L'infrastructure de MarketSpace

Est [EFSJ 97] un projet réalisé par « Telia research » et « Swedish Institute of Computer Science », il s'agit une infrastructure de commerce électronique basé sur la notion *Agent* et *Interaction*. La communication entre les agents est asynchrone et se fait par l'envoi de message en utilisant le protocole TCP-IP.

On distingue plusieurs types de messages exprimé par EOI¹¹ :

'ask' , 'tell', 'negociate', 'accept', 'refuse'.

- ✓ **ask** : pour savoir quels sont les agents intéressants.
- ✓ **tell** : pour répondre quel agent est intéressé.
- ✓ **negociate** : permet la négociation entre agents.
- ✓ **accept** : est utilisé quand un agent accepte une offre.
- ✓ **refuse** : est utilisé quand un agent refuse l'offre.

Déroulement :

1-Un agent A a besoin de savoir quel agent B est intéressé, pour cela il lui envoie un message ask(A,B,article) .

2-L'agent B qui reçoit le message répond avec la description de l'article par tell (B,A,id article).

3-L'agent A envoie un message negociate (A,B,id article,prix).

4-En fin l'agent B répond par accept ou par refuse.

9. CONCLUSION

Le commerce électronique devient de plus en plus une réalité. Or, à chaque fois qu'un marchand ou un client souhaite d'échanger des produits via l'Internet, il éprouve plusieurs heures pour trouver une bonne solution car l'information est décentralisée et elle varie indépendamment entre eux.

D'autre part, le commerce électronique est quasiment fait par nos interactions: c'est nous qui décidons quand nous allons acheter des produits ou combien nous souhaitons payer pour chaque produit, etc. Donc, la négociation joue un rôle important pendant les échanges des produits ou des services.

En résumé, il nous faut un système qui cherche les produits auxquelles nous nous intéressons (une classification selon certains critères), et qui négocie avec les différents marchands d'une manière automatique. Aujourd'hui, seules les applications agents correspondent à ce besoin.

¹¹ EOI(Expression Of Interest) est composée de :description de vendeur ou acheteur, description des articles, et les informations sur les transactions.

Chapitre III

**Proposition d'une solution
au commerce électronique
sur Internet**

1. INTRODUCTION

A fin de consacrer les chapitres de l'état de l'art, nous allons poser les problèmes liés à l'architecture classique client/serveur et aux agents mobiles. Ensuite nous allons exposer l'architecture que nous allons tenter à suivre.

2. PROBLEMATIQUE

L'architecture client/serveur est très répandue sur Internet, mais ce type de fonctionnement impose à la machine client d'être connectée en permanence durant l'interaction, et à la machine serveur des temps de services dépendant de l'état du réseau. L'approche que l'on se propose de suivre tente à réduire le problème du temps de réponse par l'intermédiaire d'agents mobiles.

Toutefois, il nous reste que le problème de sécurité. Quant un agent, cette dernière est résolue en supprimant les deux points faibles suivants :

- 1- La protection d'un site hôte contre un agent mobile.
- 2- La protection d'un agent contre un site hôte.

2.1. Protection d'un site hôte contre un agent

Un agent qui a un accès libre à un site hôte peut violer sa confidentialité, par la simple lecture, et violer son intégrité par l'écriture. La protection des sites contre des attaques menées par des agents malveillants est un problème qui est aujourd'hui bien maîtrisé. En effet, plusieurs solutions permettent maintenant de se prémunir d'éventuelles attaques et voici les méthodes les plus connues :

➤ **Bac à Sable (sandbox)**

La technique du bac à sable [CHCC 05] consiste à exécuter un agent à l'intérieur d'un environnement restreint, en interdisant l'accès au système de fichiers. Autrement dit cette technique restreint l'accès éventuel d'un code non fiable aux ressources sensibles.

Par ailleurs, il convient de distinguer deux types de classes : celles en qui on a confiance (trusted) et celles en qui on n'en a pas (untrusted).

Ainsi, un site peut exécuter un agent douteux dans le bac à sable sans trop se soucier des problèmes de sécurité.

➤ **Signature de code** : elle intervient lors de la création d'un agent, son créateur le signant numériquement afin qu'il puisse s'identifier durant ses déplacements. Cette technique permet d'obtenir une authentification de haut niveau pour les sites. Les applets Java adoptent désormais ce mode de fonctionnement. Ainsi si une applet est signée, elle est considérée comme un code de confiance. Elle sera placée dans un bac à sable dans le cas contraire [CHCC 05].

➤ **Contrôle d'accès** : il permet de mixer les deux premières techniques en offrant aux agents signés plus de fonctionnalité qu'un simple bac à sable sans pour autant accéder à toutes les fonctionnalités [CHCC 05].

➤ **Vérification du code** : elle permet d'obtenir une garantie sur la sémantique d'un code à travers l'analyse de sa structure, ou de son comportement pour un agent, en fonction d'une politique de sécurité donnée. Les bacs à sable font déjà des vérifications rudimentaires en cours d'exécution, principalement pour garantir le typage des opérandes des instructions, mais cela est fortement coûteux. Une autre approche est de vérifier automatiquement le code, avant son lancement, en s'appuyant sur une preuve de conformité [LOUR 01].

2.2. Protection d'un agent contre un site hôte

À l'opposé de la protection des sites, la protection des agents contre des sites malveillants ne dispose pas de solution éprouvée et reste encore aujourd'hui un champ de recherche ouvert. Un agent, lors de son exécution sur un site malveillant, transporte des éléments pouvant être cible d'attaque [LOUR 01]:

- ✓ **Le Code** : ensemble des instructions composant la tâche de l'agent.
- ✓ **Les données statiques** : données ne changeant pas durant les déplacements (la signature par exemple).
- ✓ **Les données collectées** : ensemble des résultats obtenus au cours des déplacements réalisés par l'agent depuis son lancement.
- ✓ **L'état courant** : ensemble de données servant à l'exécution courante de l'agent.

La sécurité des agents mobiles consiste alors à garantir les critères de confidentialité et d'intégrité de l'ensemble de ces éléments. Du point de vue des données, il est évident qu'un agent ne souhaite pas divulguer des informations critiques à n'importe quel site. Par exemple, un site malveillant pourrait récupérer la signature d'un code et l'utiliser pour créer un nouvel agent afin de s'introduire dans des environnements auxquels il n'a normalement pas accès. Pour le code, un agent transporte un savoir-faire propre à son concepteur qui pourrait tomber aux mains de ses concurrents.

Nous pouvons classer trois grandes catégories d'attaques que les sites sont capables de réaliser :

- **L'inspection** consiste à examiner le contenu de l'agent, ou le flot d'exécution afin de récupérer des informations critiques transportées par l'agent.
- **La modification** se réalise en remplaçant certains éléments de l'agent dans le but de conduire une attaque. En remplaçant le code, l'agent effectuera des opérations malveillantes sur les futurs sites à visiter.
- **Le rejeu** s'obtient en clonant l'agent puis en exécutant le clone dans plusieurs configurations pour retrouver le savoir de l'agent.

3. PROPOSITION

La solution de sécurité que nous tentant à suivre par la suite est « *d'interdire à un agent de migrer directement ou complètement vers un hôte* », et « *à un hôte d'accueillir des agents* ». Toutefois, une interaction agent-hôte doit s'effectuer, la solution est d'intervenir un site intermédiaire entre le site client et le (s) site (s) destination ; ce site accueillera les agents mobiles qui migrent et les mettra en relation avec les sites destination. C'est bien un **proxy** ou site de proximité permettant aux clients d'accéder à un site externe en exécutant leurs requêtes en son nom.

En effet, il s'agit d'une migration partielle (ou indirecte) des agents mobiles et le site proxy exécute les requêtes des agents mobiles en utilisant des invocations distantes avec les sites destination (voir figure III.1).

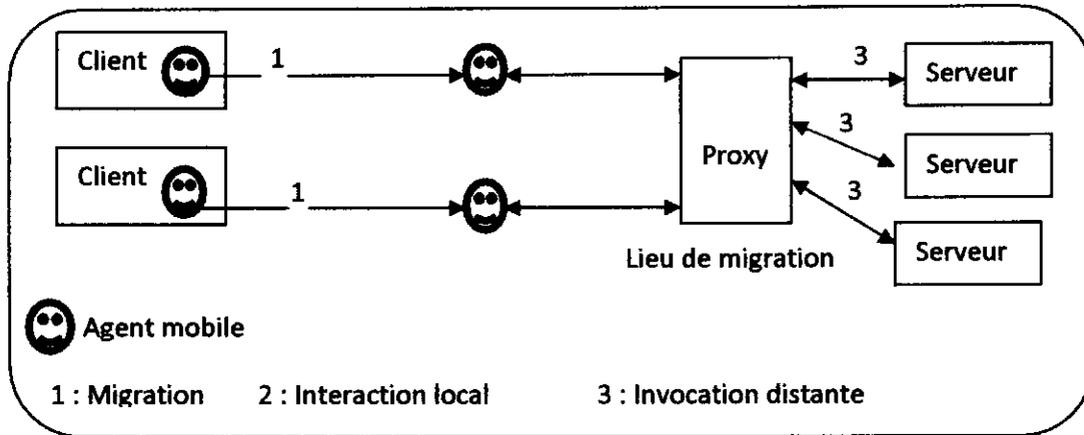


Figure III.1 Interaction des agents mobiles à travers un proxy [MNCR 04]

-L'avantage de cette approche est la réduction de problème de la sécurité multiple, il suffit donc que le site proxy soit de haute confiance.

D'autre part, pour que le service proxy qui est en fin chargé d'exécuter les requêtes des agents mobiles selon un schéma classique de client-serveur ne fait pas réduire les points forts des agents mobiles les services destinations seront implémentés par la suite comme des agents mobiles serveurs(ou vendeurs de service) .

En fin, les sites proxies seront des lieux de rencontre (dans notre cas place de marché) des agents mobiles (voire figure III.2) où ils pourront interagir localement dont nous distinguons :

- ✓ Les agents qui représentent les clients ou **agents acheteurs de services**.
- ✓ Les agents qui représentent les services ou **agents vendeurs de services**.

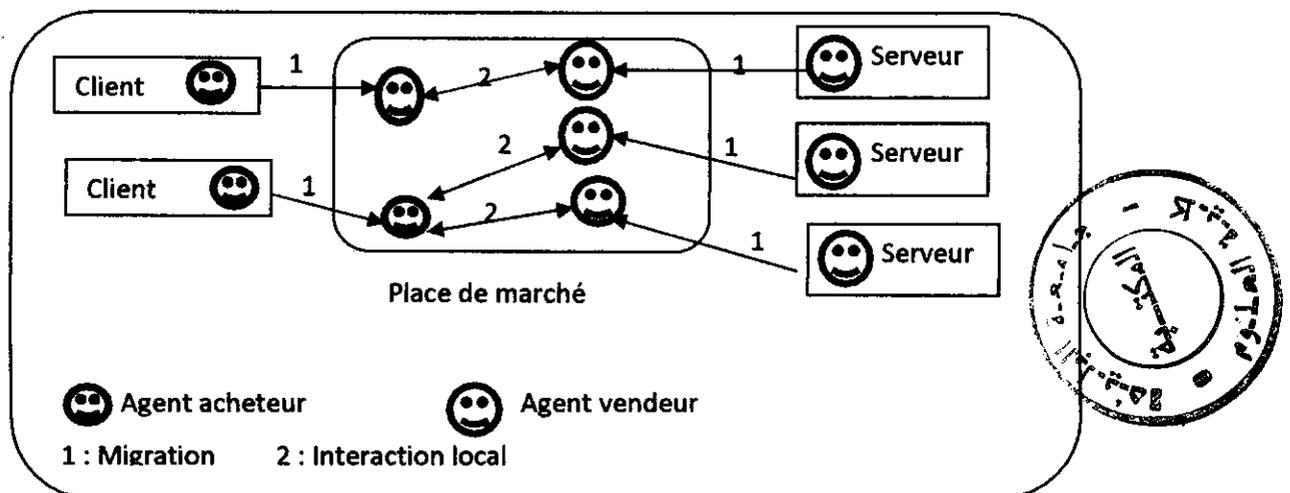


Figure III.2 Interaction des agents mobiles à travers une place de marché [MNCR 04]

En effet, la terminologie de commerce électronique sera complète en supprimant les deux problèmes posés par les agents mobiles.

Différentes manières d'appliquer le modèle Acheteur-Vendeur

Il existe trois schémas possibles d'implémentation Acheteur-Vendeur:

-**SB0 (Seller-Buyer0)** : l'acheteur et le vendeur sont statiques.

-**SB2 (Seller-Buyer2)** : l'acheteur est mobile et le vendeur est statique.

-**SB3 (Seller-Buyer3)** : l'acheteur et le vendeur sont mobiles, le lieu de rencontre est une place de marché.

Nous présentons ci-dessus les différents schémas :

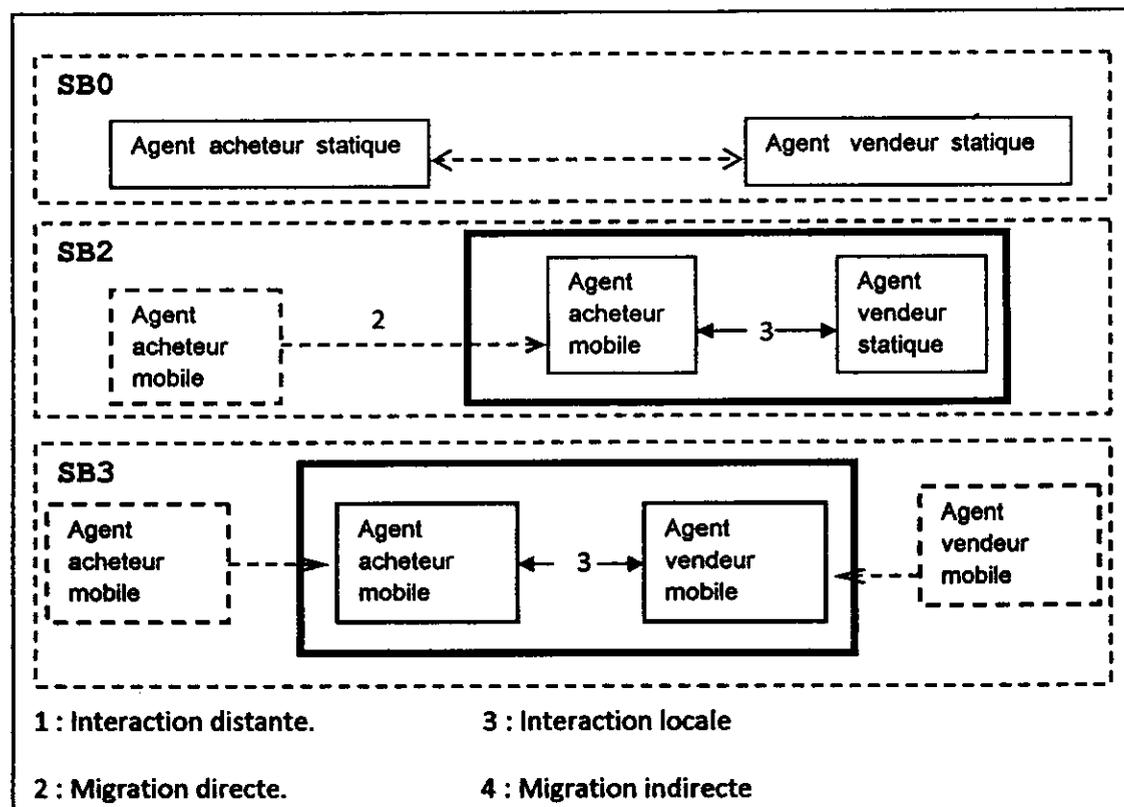


Figure III.3 Différentes implémentations du schéma SB [MNCR 04]

4. ARCHITECTURE PROPOSEE

L'architecture schématisée en page suivante que nous tentons à suivre est celle basée sur le schéma SB3

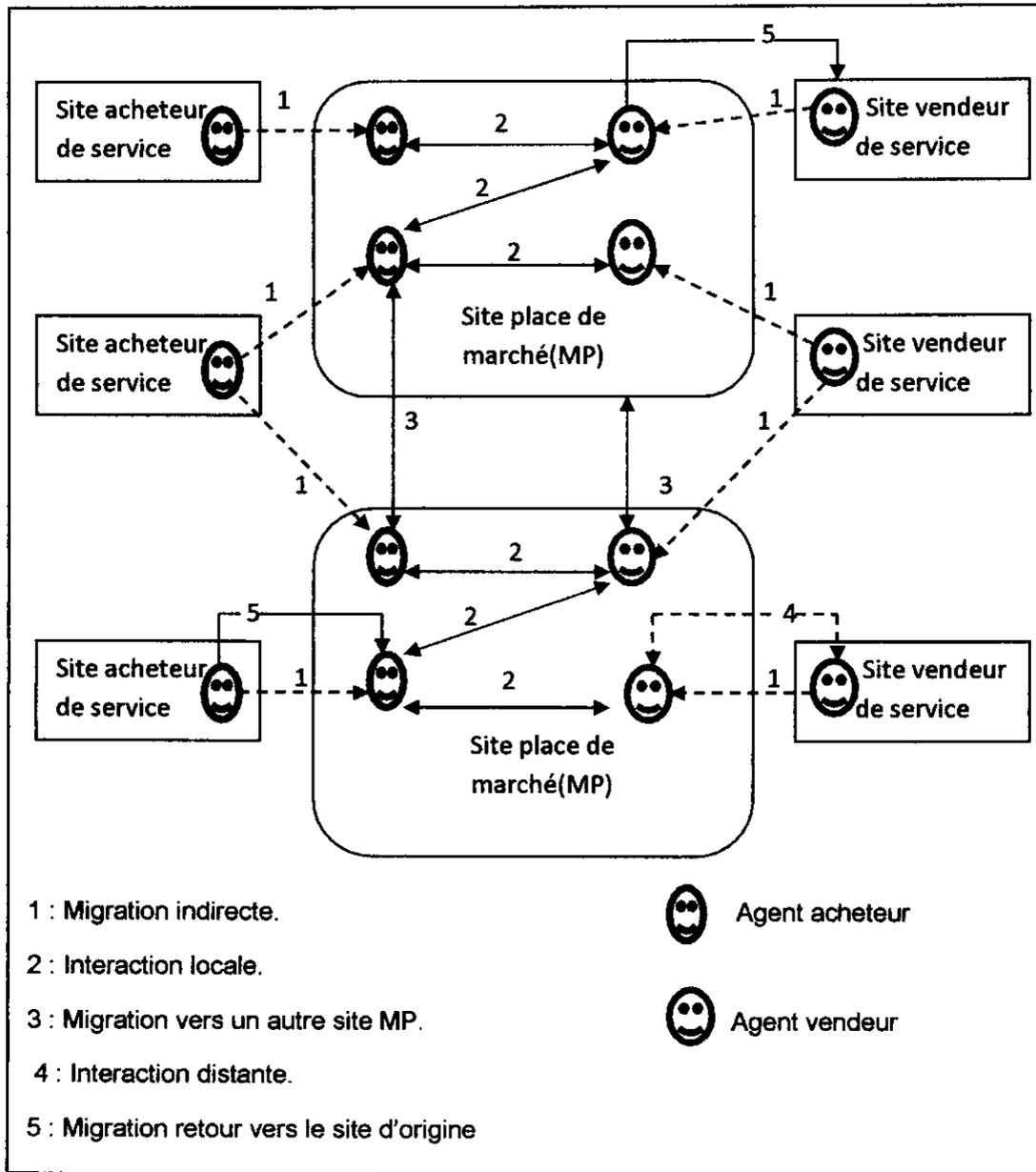


Figure III.4 Architecture proposée [MNCR 04]

4.1. Type de sites

Il existe trois types de sites :

- ✓ Les sites acheteurs de services.
- ✓ Les sites vendeurs de services.
- ✓ Les sites de rencontre des agents (places de marché).

4.2. Type d'agent

Il existe trois types d'agents :

- ✓ Les agents mobiles acheteurs (MBA).
- ✓ Les agents mobiles vendeurs (MSA).

- ✓ Les agents statiques facilitateurs(SFA), qui stimulent la rencontre entre les agents acheteurs et vendeurs sur le site MP.

4.3. Composants de l'architecture

L'architecture que nous allons suivre dans notre travail est basée sur les composants suivants :

- Acheteurs de services **CBS** (Client Buyer of services).
- Vendeurs de services **PSS** (Provider Seller of Services).
- Place marché virtuelle **MP** qui sera détaillée par la suite.
- Autorité de sécurité **TSA** (Trust & Authority Services) : gère et génère des certificats pour les MP, e-shops et les ASP, en utilisant la technologie PKI (Public Key Infrastructure) qui est basée sur le cryptage, pour cela nous avons utilisé l'algorithme **RSA**. Le TSA sert à garantir la sécurité des agents (acheteurs et vendeurs) et des places marchés visités.
- Fournisseurs de services agent **ASP** (Agent Service Provider) : est un environnement d'exécution pour les agents mobiles où se créent les agents (acheteurs ou vendeurs) pour les clients abonnés à l'ASP.
- Services de nommage **MPNS** (MP Naming Services) : offrent des informations sur les MP et leurs e-shops à la manière de DNS.

Le schéma ci-dessus représente un environnement d'Internet pour notre architecture :

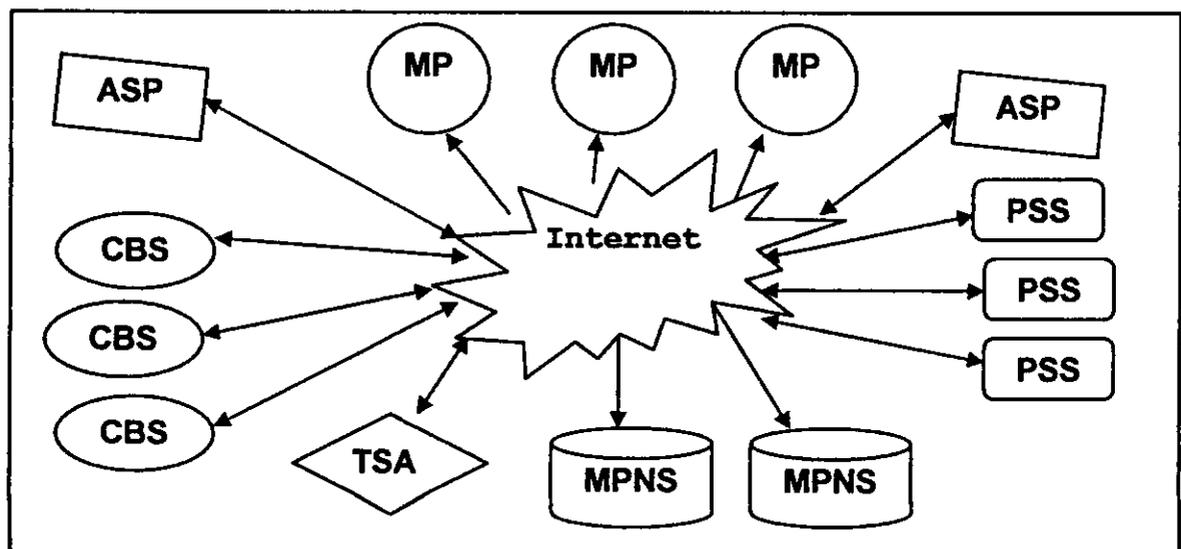


Figure III.5 Environnement Internet pour notre architecture [MNCR 04]

5. REPRESENTATION PHYSIQUE D'UNE MP

Une place de marché (MP) est un Intranet¹ où nous trouvons :

- **MPM (MP Manager)** : gère la place marché et contrôle la base de donnée d'annuaire(MPDS), qui regroupe les adresses IPs, les e-shops.
- **MPSM (MP Security Manager)** : gère la sécurité de la place marché, reçoit des certificats de visiteurs. Il agit comme un Pare-feu(Firewall) du réseau Intranet.
- **MP server** : est un serveur où MPSM et MPM sont exécutés, ainsi est un environnement recueille les agents mobiles.
- **Les e-shops** : est un serveur où les e-shops sont situés, il utilise un **service agent** (ou agent facilitateur). Le rôle d'un agent facilitateur est :
 - ✓ Sauvegarde les informations dans les bases de données de l'e-shops.
 - ✓ Faire la mise à jour des bases de données.
 - ✓ La communication avec l'agent consommateur.
 - ✓ La protection des ressources locales en e-shop.
 - ✓ Sauvegarde les e-shops en MPM.
 - ✓ L'utilisation du certificat de l'e-shop par TSA et l'envoi au MPM.

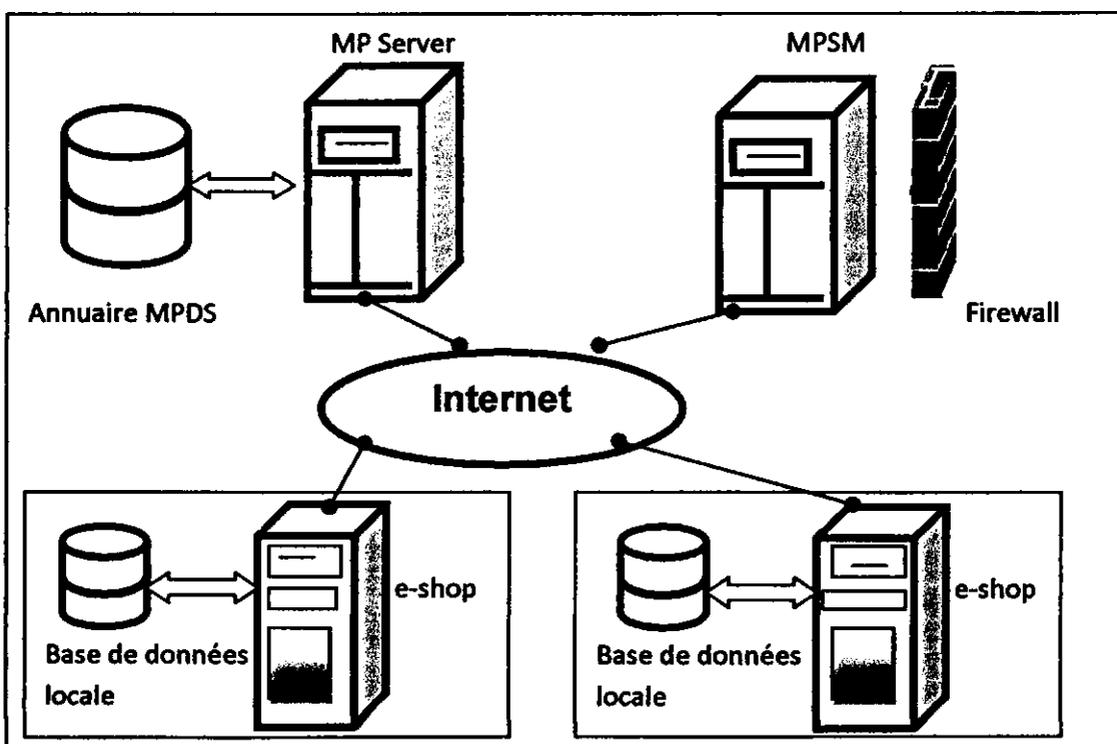


Figure III.6 Schéma physique d'une place de marché [MNCR 04]

¹ Intranet est réseau local de plusieurs hôtes

6. LES SERVICES S'EXECUTEANT DANS UNE PLACE DE MARCHÉ

Les services dans une place marchée peuvent être implémentés sous forme d'agents statiques, nous avons :

- Le service MP : est un agent d'accueil des agents, qui utilise le service MPDS.
- Le Firewall ou MPSM est un service classique qui filtre les agents incidents.
- Un agent facilitateur principal sur chaque MP qui permet de diriger les agents incidents sur la MP vers les bons e-shops.
- Un agent facilitateur local sur chaque e-shop qui permet de faciliter la rencontre des agents acheteurs et vendeurs.
- Les agents vendeurs.
- Les agents acheteurs

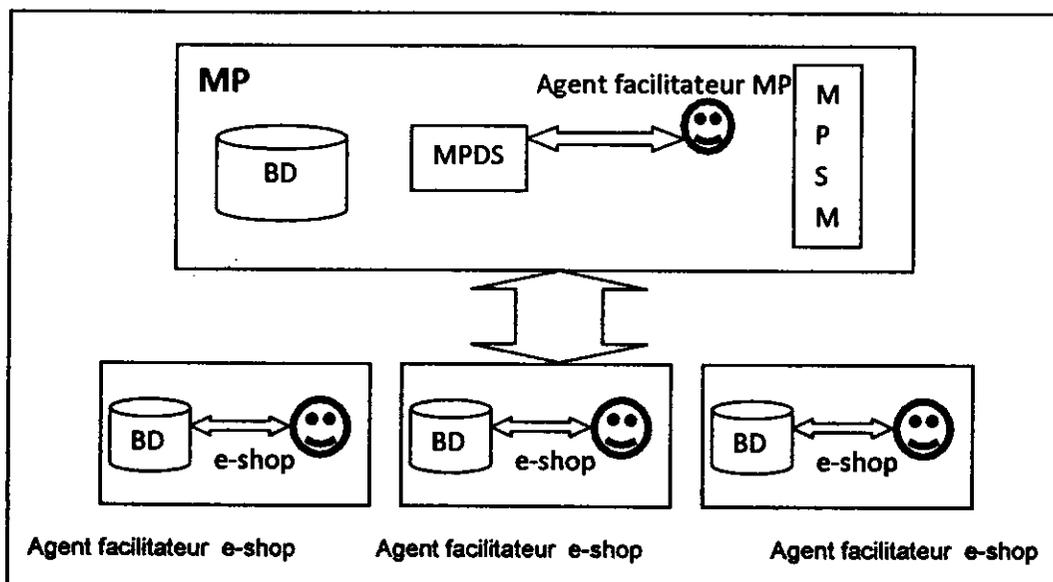


Figure III.7 schéma logique d'une place de marché. [MNCR 04]

7. BASES DE CONNAISSANCES

Notre architecture définit trois niveaux de base de connaissances qui facilitent la localisation des services et ressources nécessaires pour les agents.

7.1. La base MPNS : stockée sur les serveurs MPNS, cette base stocke les informations suivantes :

- Les ID des places de marchés (par exemple un numéro de sous réseau IP)

- Les catalogues résumés des services de chaque MP, actualisée régulièrement par les agents MPM sur les serveurs MPNS.
- La charge, la spécialisation, la promesse de chaque MP.

7.2. La base d'annuaire (MPDS) stockée sur chaque serveur MP, elle regroupe les informations de sécurité sur toutes les ressources contenues dans cet serveur, gérée et contrôlée par l'agent MPM, elle contient les informations suivantes :

- IDs et adresses IP des e-shops de la MP.
- Le catalogue de domaine contient les informations sur les services et ressources disponibles sur tous les e-shops.

7.3. La base d'e-shop : contient toutes les informations détaillées sur tous les services et ressources disponibles au niveau de l'e-shop.

7.4. Flux de données entre les différentes bases de connaissances

Les mises des trois bases précédentes sont effectuées dans un ordre chronologique du plus détaillé vers le moins détaillé.

1. Les agents vendeurs et acheteurs réalisent des transactions dans un e-shop et mettent à jour *la base locale* sous le contrôle de *l'agent facilitateur local*.
2. L'agent facilitateur d'un e-shop émet des mises à jour la base d'annuaire de la MP sous le contrôle de l'agent facilitateur principal.
3. L'agent facilitateur principal de chaque MP émet des mises à jour vers les serveurs MPNS, qui tiennent des bases de données globales sur le système entier.

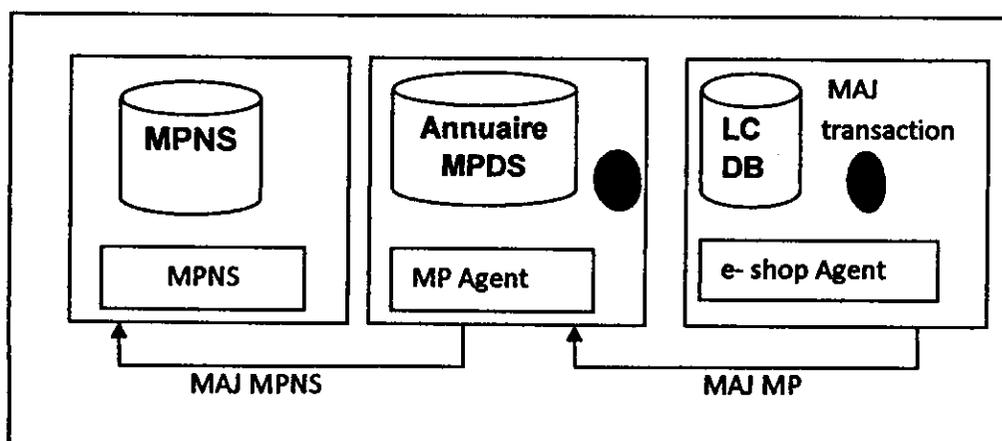


Figure III.8 Différentes bases de connaissances et flux correspondants. [MNCR 04]

8. LES ENVIRONNEMENTS ASP (Agent Services Provider)

Les ASP sont des sites qui offrent un environnement de création, de gestion et d'exécution pour les agents mobiles requis par les clients.

L'ASP est structuré par quatre modules principaux :

- Un module **serveur pour les clients**, auquel les utilisateurs se connectent via Internet.
- Un module de **création des agents mobiles**, qui crée un agent mobile pour réaliser la requête du client.
- Un module de **gestionnaire des agents**, qui permet d'inscrire les agents mobiles, leurs tâches, suivre leur migration. Ce module utilise une base de données pour stocker les informations sur les agents.
- Un module **serveur d'agent**, qui offre un environnement d'exécution des agents mobiles et facilitateurs.

9. LA SECURITE DU SYSTEME

9.1. Les problèmes posés

L'utilisation des agents mobiles dans un réseau pose des problèmes de sécurité ; dans notre système, la sécurité porte les problèmes suivants:

- **Intégrité des requêtes** : la requête transportée par un agent mobile ne doit pas être modifiée.
- **Intégrité des résultats** : les résultats ramenés par un agent doivent refléter la réalité.
- **Confidentialité** : l'information collectée par un agent au cours de son itinéraire ne doit pas être révélée à un autre agent ou un hôte.
- **Authenticité** : l'agent mobile doit suivre l'itinéraire qui lui a été communiqué par un serveur MPNS.

9.2. Les solutions possibles

L'approche que nous allons développer utilise les techniques de cryptographie. De plus nous trouvons les solutions suivantes pour éviter les problèmes précédents de sécurité :

➤ **Un réseau fermé** constitué par les places de marché, les serveurs MPNS, les sites ASP. Ce réseau est ouvert à tous les sites à condition qu'ils s'inscrivent auprès d'un ASP, et respectent un certain nombre de règles. Pour que ces sites puissent s'inscrire, nous avons ajouté un élément clé TSA ou autorité de sécurité et de confiance, qui fournit des certificats pour les agents mobiles et les hôtes. Cette certificat contient l'identité du site ou l'utilisateur, sa clé privée, sa clé public. Ces clés sont utilisées pour crypter les données entre un émetteur et un récepteur selon l'algorithme RSA.

➤ **La prévention de la falsification des agents**

Pour protéger l'intégrité de la requête transportée par un agent mobile, ce dernier peut signer sa requête avec sa clé privée de l'agent. Lorsqu'un agent visite un hôte, celui-ci peut contrôler la requête en obtenant la clé publique de l'agent auprès du TSA. De plus, l'agent qui répond à la requête peut crypter les résultats avec la clé publique de premier agent.

➤ **La détection de la falsification des agents**

Les hôtes visités ou d'autres agents peuvent modifier l'itinéraire d'un agent mobile, pour protéger l'itinéraire, une solution consiste à laisser l'itinéraire en claire, i.e. un hôte H1 peut alors le lire puis le crypter avec sa clé privée, en formant la chaîne suivante :

$-I0 \rightarrow I1 = I0 + E11$ où $I0$ est l'itinéraire d'origine non crypté, et $I1$ l'itinéraire $I0$ crypté par le hôte H1, et + la concaténation des deux chaînes.

-L'hôte suivant exécute le même processus, ainsi de suite. A l'arrivée, l'ASP va décrypter les itinéraires en utilisant les clés publiques des hôtes visités. Si un hôte malicieux a falsifié l'itinéraire, ce sera détecté.

➤ **Protection des MP**

Chaque MP dispose d'un pare-feu MPSM qui filtre les agents incidents en contrôlant les certificats présentés par ces derniers. Les MP se comportent comme des réseaux privés, il reçoit les informations de sécurité des agents qui se préparent à y migrer. Ces informations sont émises par les ASP ou la TSA, selon l'itinéraire de l'agent.

Le schéma suivant résume notre modèle d'architecture

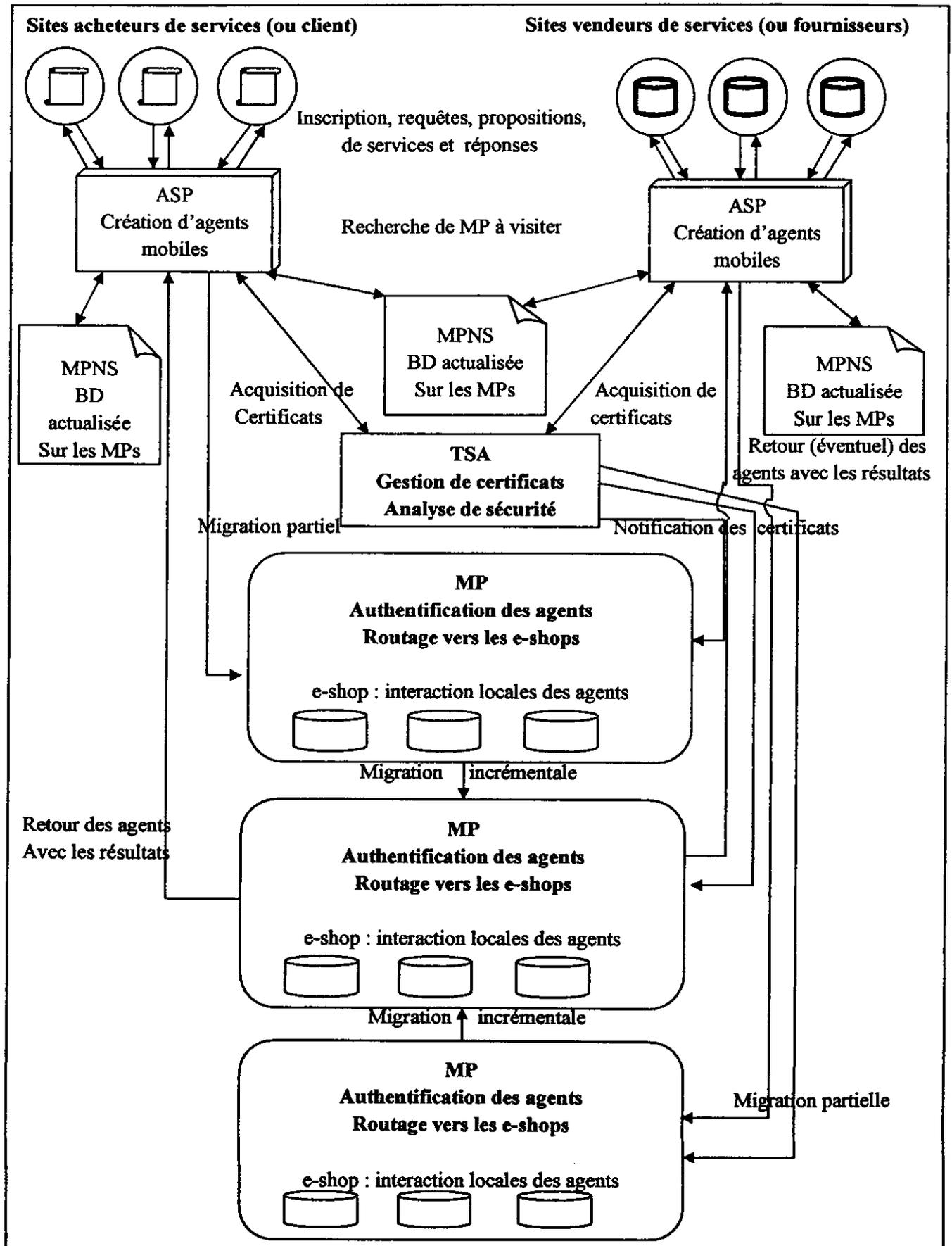


Figure III.9 vue générale de notre architecture proposée [MNCR 04]

10. CONCLUSION

Dans ce chapitre, nous avons montré les différents problèmes coïncidant les pouvoirs des agents dans le cadre d'amender les places de marché virtuelles. Par conséquent, nous avons détaillé l'architecture à suivre pour les échapper.

Chapitre IV

Conception du système

1. INTRODUCTION

Dans ce chapitre, nous allons utiliser le langage UML pour étudier un modèle basé sur les notions du commerce électronique, concernant la **recherche et vente des formations sur Internet**, en utilisant une place de marché à base d'agents mobiles sécurisés (MPSMA¹).

2. DESCRIPTION DU MODELE DE COMMERCE ELECTRONIQUE

Le commerce électronique possède plusieurs applications, y compris l'application de **la recherche et vente des formations sur Internet** qui est une application répartie. Cette application a pour objectifs :

- ✓ Rechercher une formation
- ✓ La Négociation des formations
- ✓ Commander des formations.
- ✓ Offrir des formations.

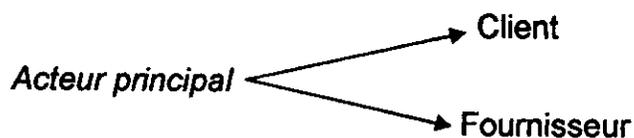
Cette application du commerce électronique se compose de :

- Un ou plusieurs sites d'utilisateurs (clients ou fournisseurs).
- Un ou plusieurs sites de commerce électronique.

Pour mieux comprendre notre modèle de commerce électronique, nous allons le modéliser sous une forme de *cas d'utilisations*, ce qui permet de suivre un critère de décomposition fonctionnel, dont les besoins peuvent être exprimés.

Un **acteur** représente un rôle joué par une personne externe, ou un processus qui interagit avec un système, dans notre système nous distinguons deux types d'acteurs :

- ✓ **Acteur Principal** regroupe les personnes qui utilisent les fonctions principales du système. Dans notre cas, il s'agit des clients et des fournisseurs



- ✓ **Autres système** regroupe les systèmes avec lesquels le système doit interagir. Dans notre cas, il s'agit une *banque* pour effectuer les transactions financières.

¹ MPSMA: Market Place with Security Mobile Agent.

Donc nous pouvons distinguer les cas suivant :

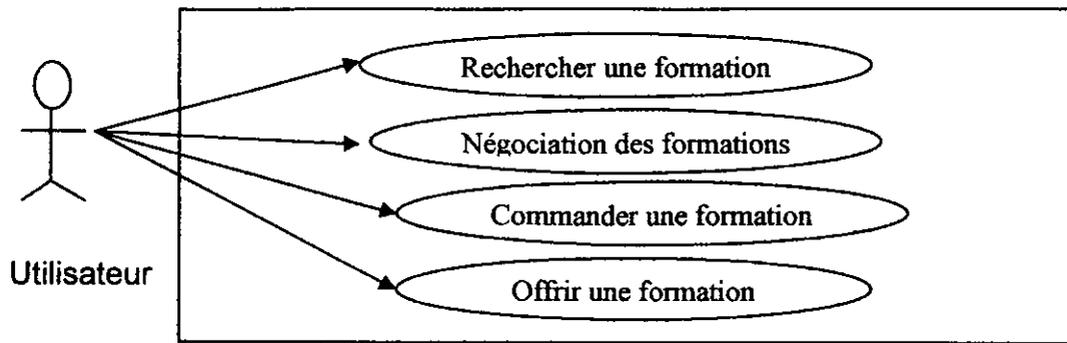


Figure IV.1 diagramme de cas d'utilisation générale de l'application de commerce électronique.

2.1 Rechercher une formation

-Une fois qu'un client se connecte sur notre site, il trouve un formulaire pour identifier ses besoins. Les informations saisies sont:

L'intitulé de formation, type, domaine, Date début, date fin, durée, le lieu, prix min, prix max

- Le client doit recevoir une réponse qui satisfait ses besoins.

Pour que ces deux tâches soient réalisées, nous devons découper la fonction *Recherche une formation* en deux sous fonctions :

- 1- Identification des besoins.
- 2- vérification de l'existence d'une formation.

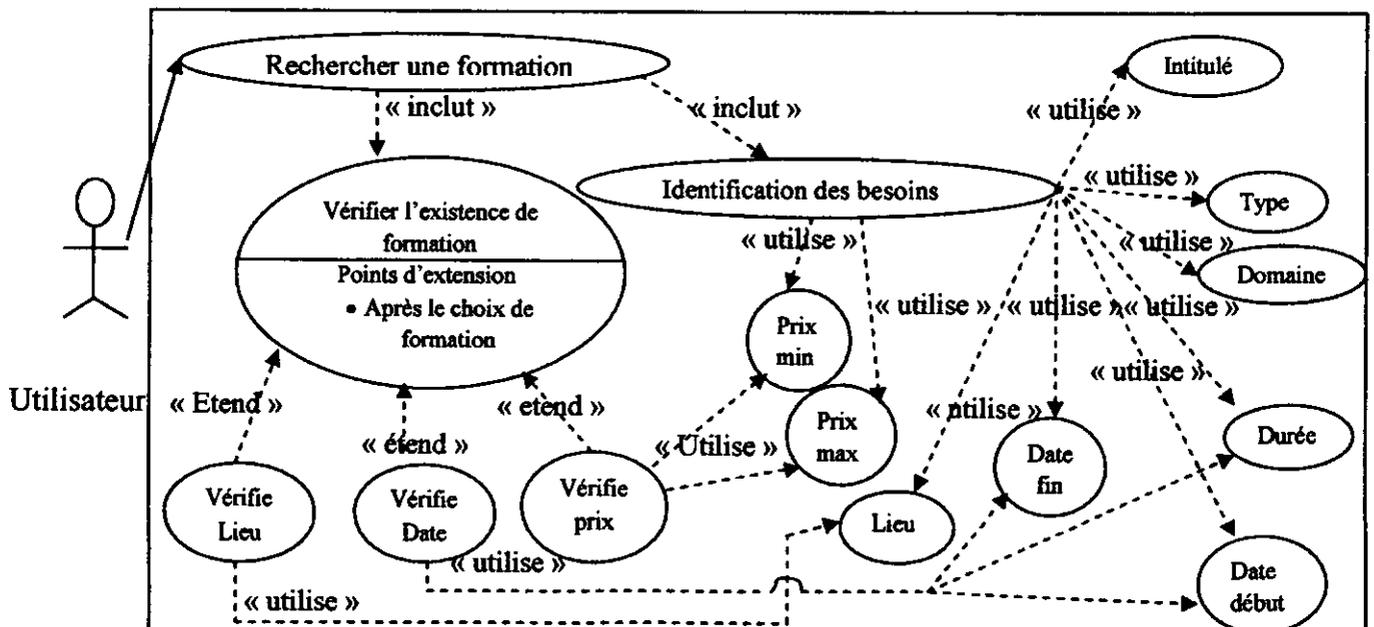


Figure IV.2 diagramme de cas d'utilisation de recherche une formation

2.2. Négociation

Après la recherche d'une formation, Le processus de négociation est défini selon trois aspects :

1-La négociation de prix tel que $\text{prix min} \leq \text{prix final} \leq \text{prix max}$.

2-La négociation de date.

3-La négociation de lieu tel que lieu=lieu de formation ou un lieu plus proche.

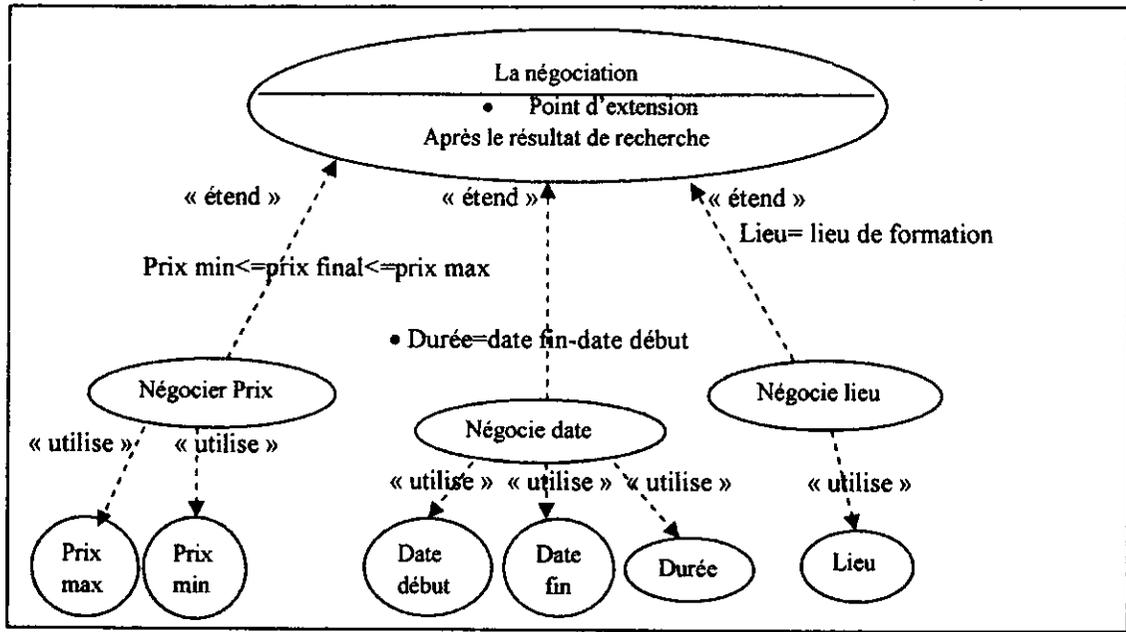


Figure IV.3 diagramme de cas d'utilisation de la Négociation

2.3. Commander une formation

Après une négociation, une réponse est envoyée contenant les résultats. Si le client est d'accord avec ses résultats, il identifie ses informations bancaires (nom, prénom, numéro de carte de crédit, numéro de compte bancaire, date expiration), et valide la commande par l'envoi *un bon de commande* au vendeur.

Enfin le client effectue un paiement pour valider la transaction.

Si le client n'est pas d'accord, annule la transaction.

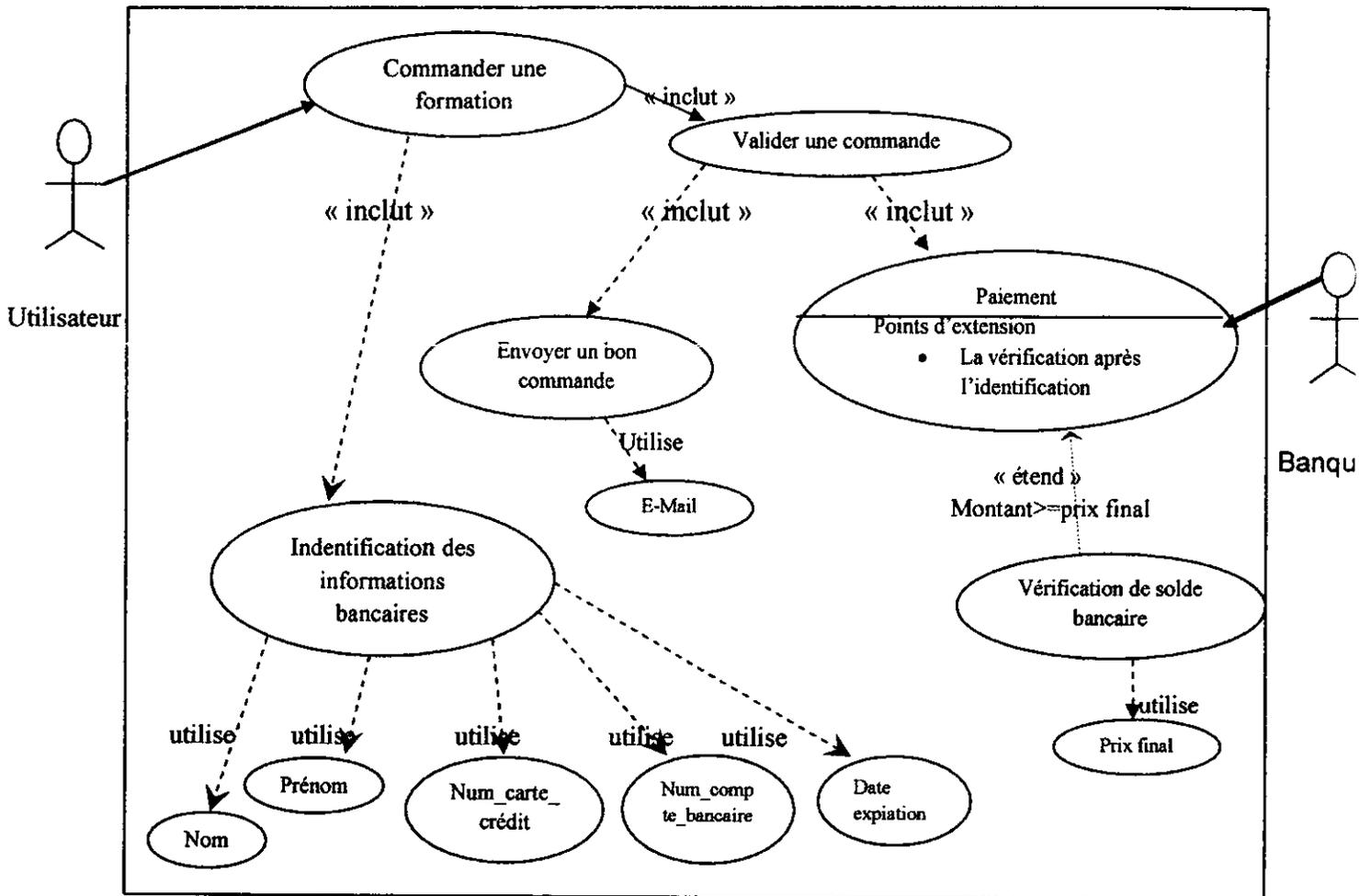


Figure IV.4 diagramme de cas d'utilisation de commander une formation

2.4. Offrir une formation

Un fournisseur peut offrir des formations comme suit :

- Une fois qu'un fournisseur se connecte sur notre site, il propose ses différentes formations à vendre.
- Le fournisseur saisit ses informations nécessaires dans un formulaire. Nous distinguons les informations appropriées à la formation proposée (intitulé de formation, type, domaine, Date début, date fin, durée, le lieu, prix min, prix max) ; et les informations appropriées au fournisseur (identifiant de fournisseur, Adresse e-mail, Numéro Tel, Numéro de fax, Adresse).
- Le fournisseur reçoit un message indiquant que son inscription est réussie.

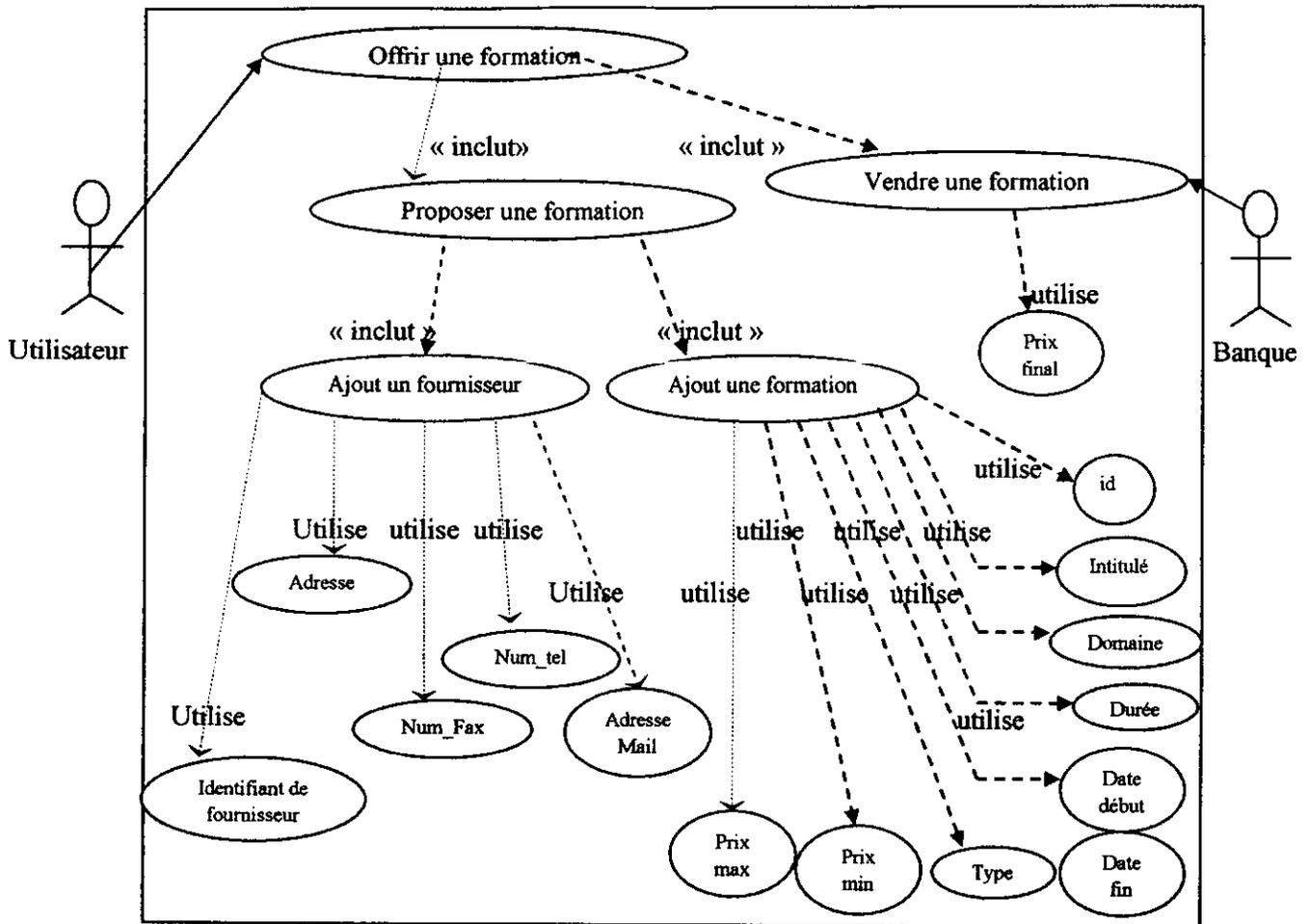
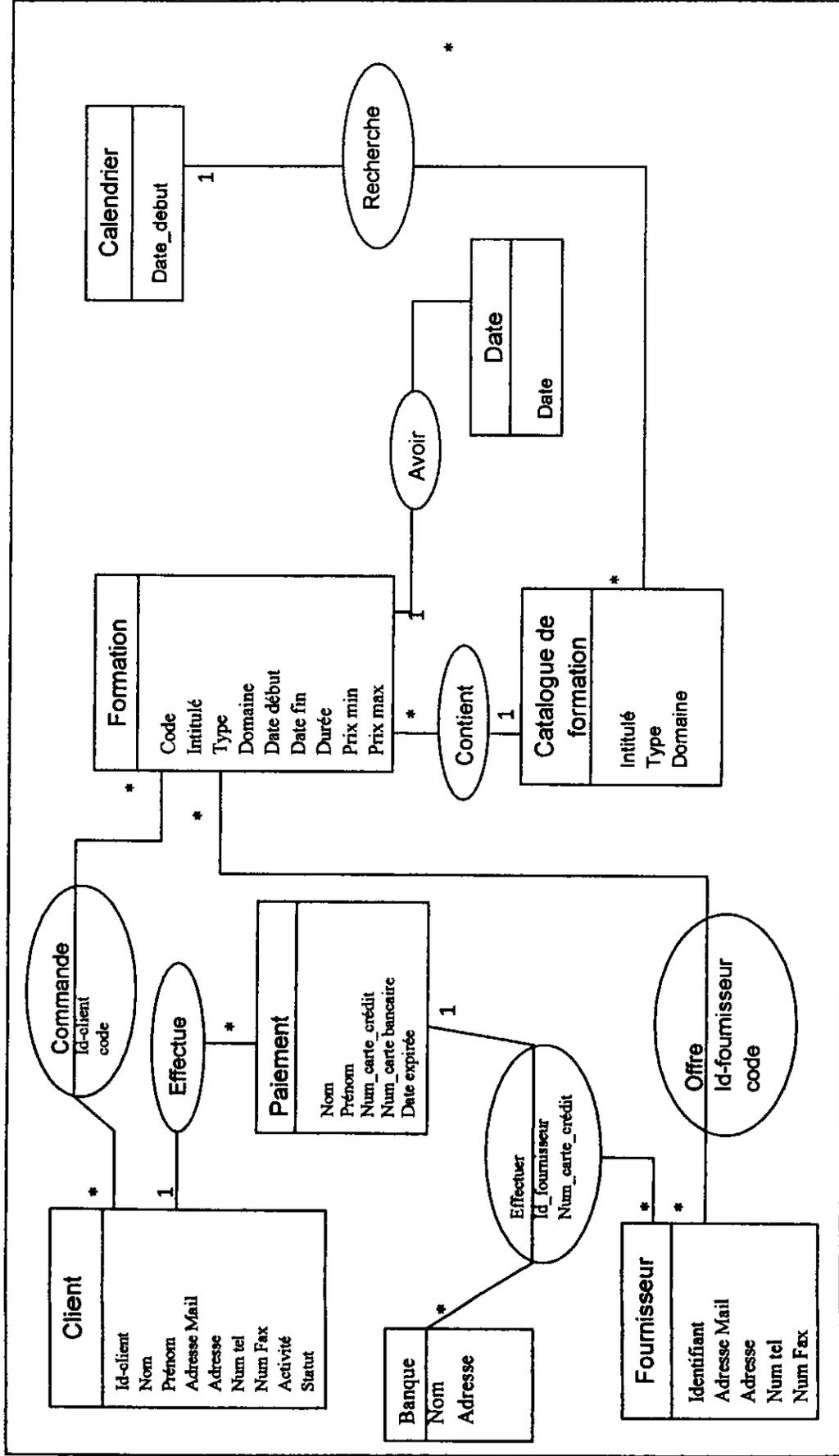


Figure IV.5 diagramme de cas d'utilisation offrir une formation

2.5. Diagramme de cas d'utilisation détaillé du modèle de commerce électronique « la recherche et vente des formations sur Internet »

Nous allons détailler maintenant notre modèle de commerce électronique

2.6. Diagramme de classe de modèle de commerce électronique



3. DESCRIPTION DU SYSTEME MPSMA

Maintenant nous allons appliquer le modèle du commerce électronique (la recherche et vente des formations sur Internet) à l'architecture proposée en chapitre III.

Notre modèle est basé sur des places de marché où s'effectuent les transactions commerciales relatif à recherche et vente des formations sur Internet, où nous utilisons des agents mobiles sécurisés (BMA² et SMA³), et des agents facilitateurs (ASP_IFA, MP_IFA, e-shop_IFA).

Chaque place de marché contenant MPSMA est un site web du commerce électronique qui représente un éditeur (Microsoft, Oracle, etc.) où :

- Les fournisseurs peuvent proposer leurs formations.
- Les clients peuvent trouver leurs besoins.

Le Tableau en bas représente un exemple des MPs de MPSMA :

Tableau IV.1 Exemple des places de marché dans MPSMA.

MP1	Microsoft.com
MP2	Hp.com
MP3	Oracle.com
MP4	CISCO.com

Dans notre système, les transactions financières s'effectuent par un VBA⁴, ce dernier génère un VAN⁵ pour chaque utilisateur inscrit sur l'ASP. Ce VAN est initialisé avec un certain montant.

En effet, une application de commerce électronique est une application répartie qui se compose de :

- Un ou plusieurs sites d'utilisateurs (clients ou fournisseurs).
- Un ou plusieurs sites de place de marché (MP).

² BMA: Buyer Mobile Agent.

³ SMA: Seller Mobile Agent.

⁴ VBA: Virtual Banking Authority.

⁵ VAN : Virtual Account Number

3.1. Diagramme de cas d'utilisation

Comme nous avons décrit auparavant qu'il y a deux acteurs, l'un est principal et autre système, dans notre modèle, le premier acteur est l'utilisateur (client ou fournisseur), et le deuxième est VBA (Virtual Banc Authority).

3.1.1. Commander une formation : le schéma suivant montre les différentes sous fonctions de *Commander une formation* :

3.1.2 Offrir une formation: le schéma suivant montre les différentes sous fonctions d'offrir une formation

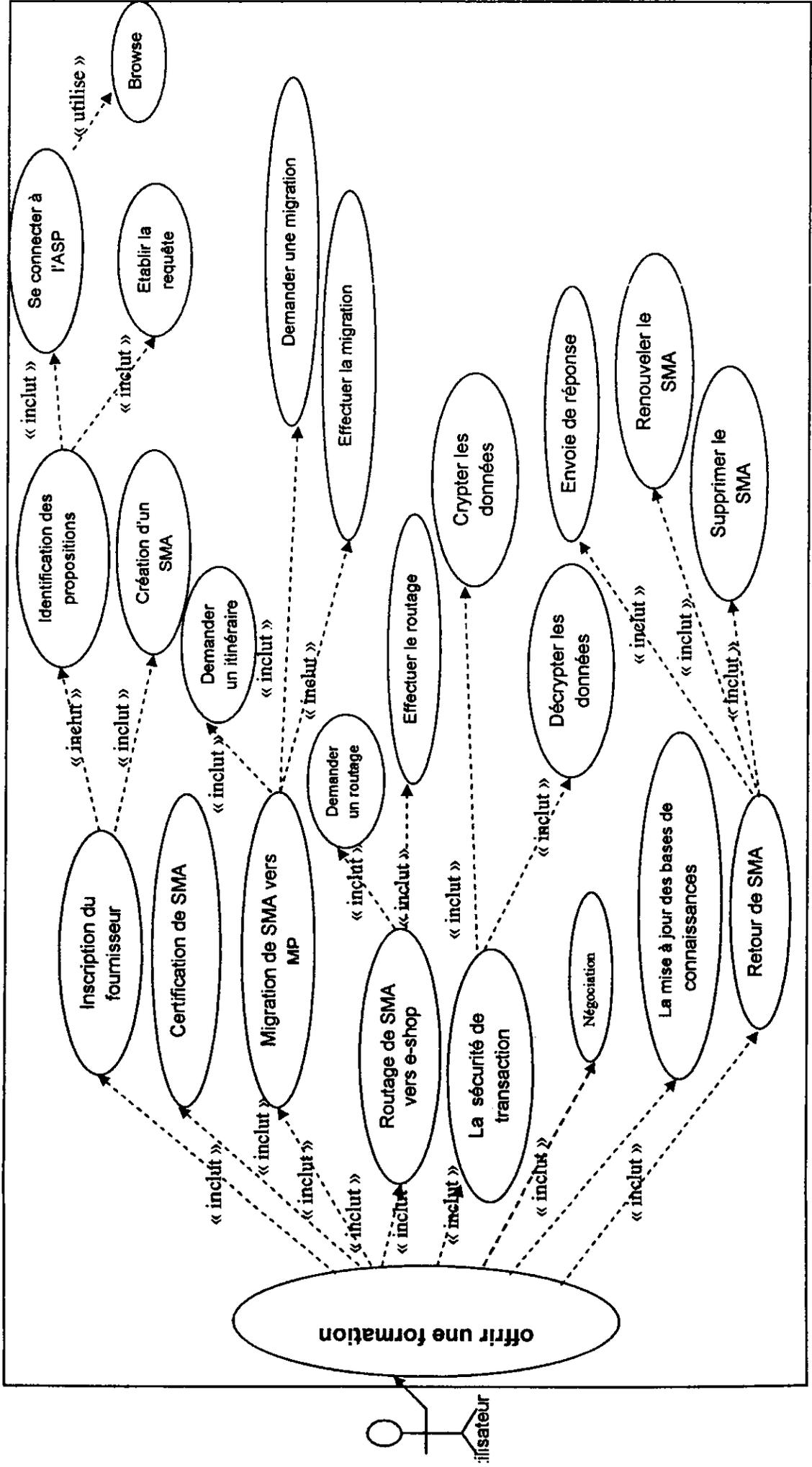


Figure IV.8 Décomposition de fonction offrir une formation en sous fonctions

3.2. SCENARIOS ET DIAGRAMMES DE SEQUENCES

3.2.1. Commander une formation

Scénario1 : Identification des besoins

- 1- Un client entre le browser de site MP, et connecte à ASP.
- 2- ASP lui envoie une applet (formulaire).
- 3- Le client remplit cette applet (formulaire) en spécifiant ses besoins (identifiant de formation, intitulé, type, domaine, date début, date fin, durée, prix min, prix max).
- 4- Le client envoie cette applet (formulaire) à l'ASP.

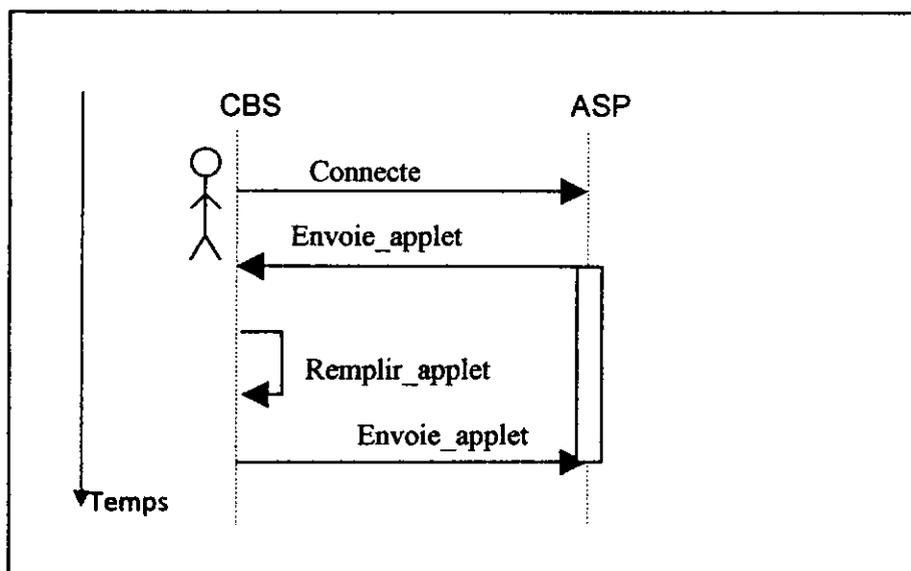


Figure IV.9 diagramme de séquence de l'inscription d'un client.

Scénario2 : création d'un compte VAN

- 1- ASP demande au client d'envoyer ses informations bancaires.
- 2- Le client crypte ses informations bancaires et les envoie à ASP.
- 3- ASP envoie à VBA le nom complet du client et la clé publique de VBA(Pv).
- 4- VBA envoie à l'ASP un numéro de référence (num_ref) avec clé secrète du VBA(Sv) et la clé publique du client (Pu).
- 5- ASP envoie à VBA le numéro de référence du client (num_ref) avec son numéro de carte de crédit (compte_cc) et son compte bancaire (CB) et clé publique du client (Pu) et du VBA (Pv).

6- VBA envoie à ASP un numéro de compte virtuel VAN et la clé publique du client (Pu).

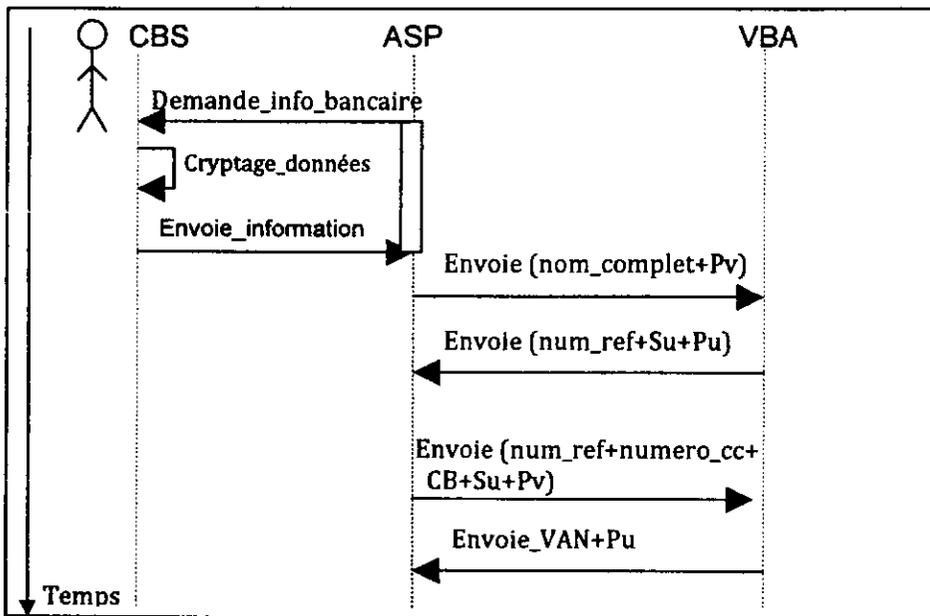


Figure IV.10 diagramme de séquence de création d'un compte VAN

Scénario3 : création d'un BMA

- 1- Le client qui déjà identifié ses besoins et qui a un compte VAN, demande à l'ASP de créer un agent mobile acheteur (BMA).
- 2- L'ASP crée le BMA en utilisant un identifiant, la requête et le compte VAN.
- 3- L'ASP notifié le client de la création de BMA.
- 4- Dans ce cas le client est inscrit et il peut se déconnecter.

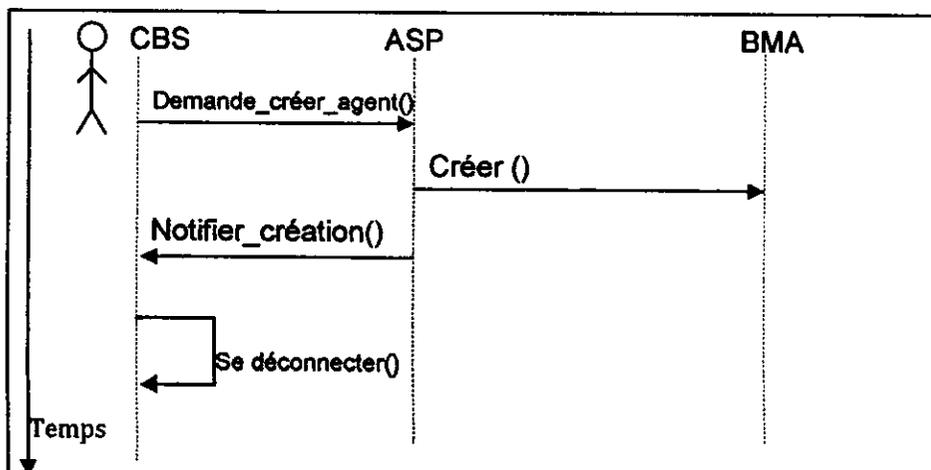


Figure IV.11 diagramme de séquence de création d'un BMA

Scénario 4 : Certification de BMA

- 1- BMA demande à l'ASP_IFA de lui affecter un itinéraire initial.
- 2- ASP_IFA envoie une requête de demande d'itinéraire au MPNS.
- 3- MPNS propose un itinéraire, et il l'envoie à l'ASP_IFA.
- 4- ASP_IFA affecte l'itinéraire au BMA.
- 5- BMA demande à l'ASP_IFA d'affecter un certificat.
- 6- ASP_IFA envoie au TSA la requête et l'itinéraire, et lui demande de certifier l'agent (il crypte les données précédentes avec une clé publique et une clé privée).
- 7- TSA créer un certificat pour le BMA.
- 8- TSA envoie ce certificat à l'ASP_IFA.
- 9- ASP affecte ce certificat au BMA.
- 10-TSA sauvegarde ce certificat dans sa base de connaissances.

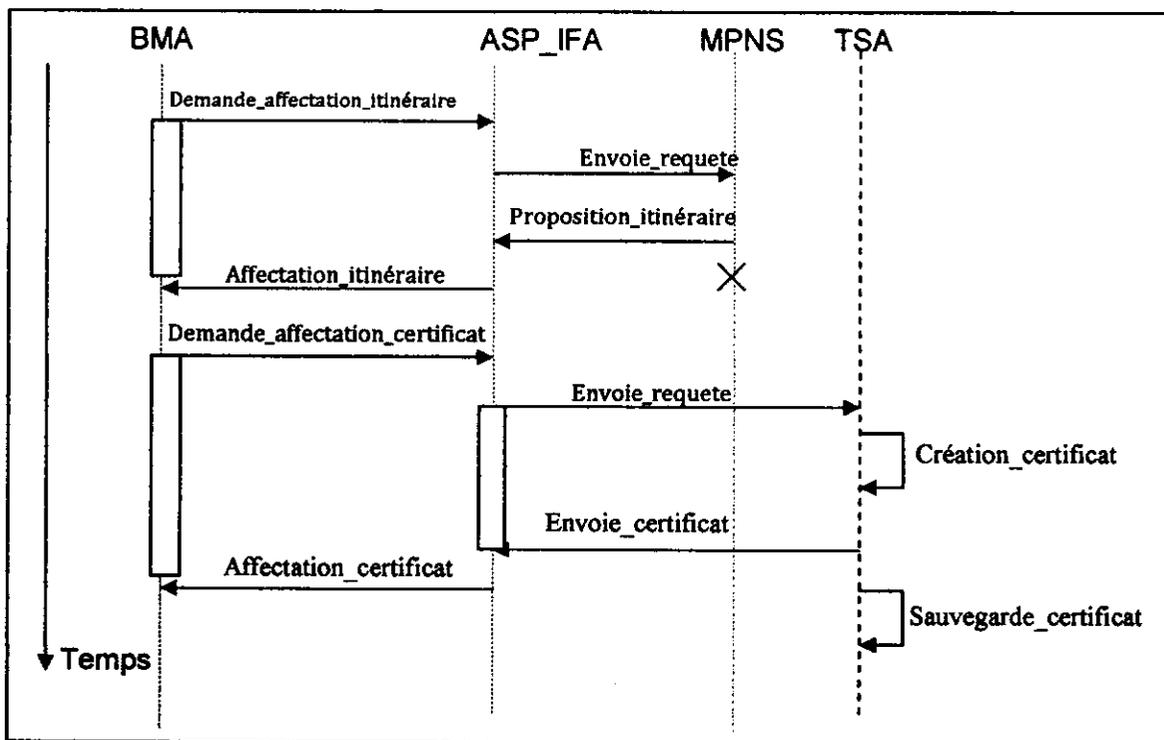


Figure IV.12 diagramme de séquence de certification de BMA.

Scénario5: migration de BMA vers les MPs

- 1- BMA demande une autorisation de migration à MP_IFA1 de site MP1 de son itinéraire.
- 2- MP_IFA1 acquiesce la demande de BMA.
- 3- BMA notifié sa migration à ASP_IFA.
- 4- Au niveau de site MP1, BMA demande d'être authentifié et contrôlé par le service MPSM1.
- 5- Le service MPSM1 demande un jeton au MPDS1 du site MP1.
- 6- MPDS1 envoie ce jeton au MPSM1, ce jeton permet d'accepter ou refuser le BMA.
- 7- MPSM1 affecte ce jeton au BMA.
- 8- BMA authentifié demande à MP_IFA1 de le router vers les hôtes d'e-shops.
- 9- MP_IFA1 consulte sa base (annuaire) MPDS1 pour vérifier l'existence de formation commandée.
- 10- Si la formation est existe, l'annuaire MPDS1 envoie à MP_IFA1 un message «formation existante».
- 11- Sinon MPDS1 envoie à MP_IFA1 un message «formation non existante »
BMA va migre vers un autre site MP2.
- 12- Tant que la formation commandée ne soit pas encore trouvée, le processus de migration sera répété sur le reste des sites MP présents dans l'itinéraire, l'un après l'autre.

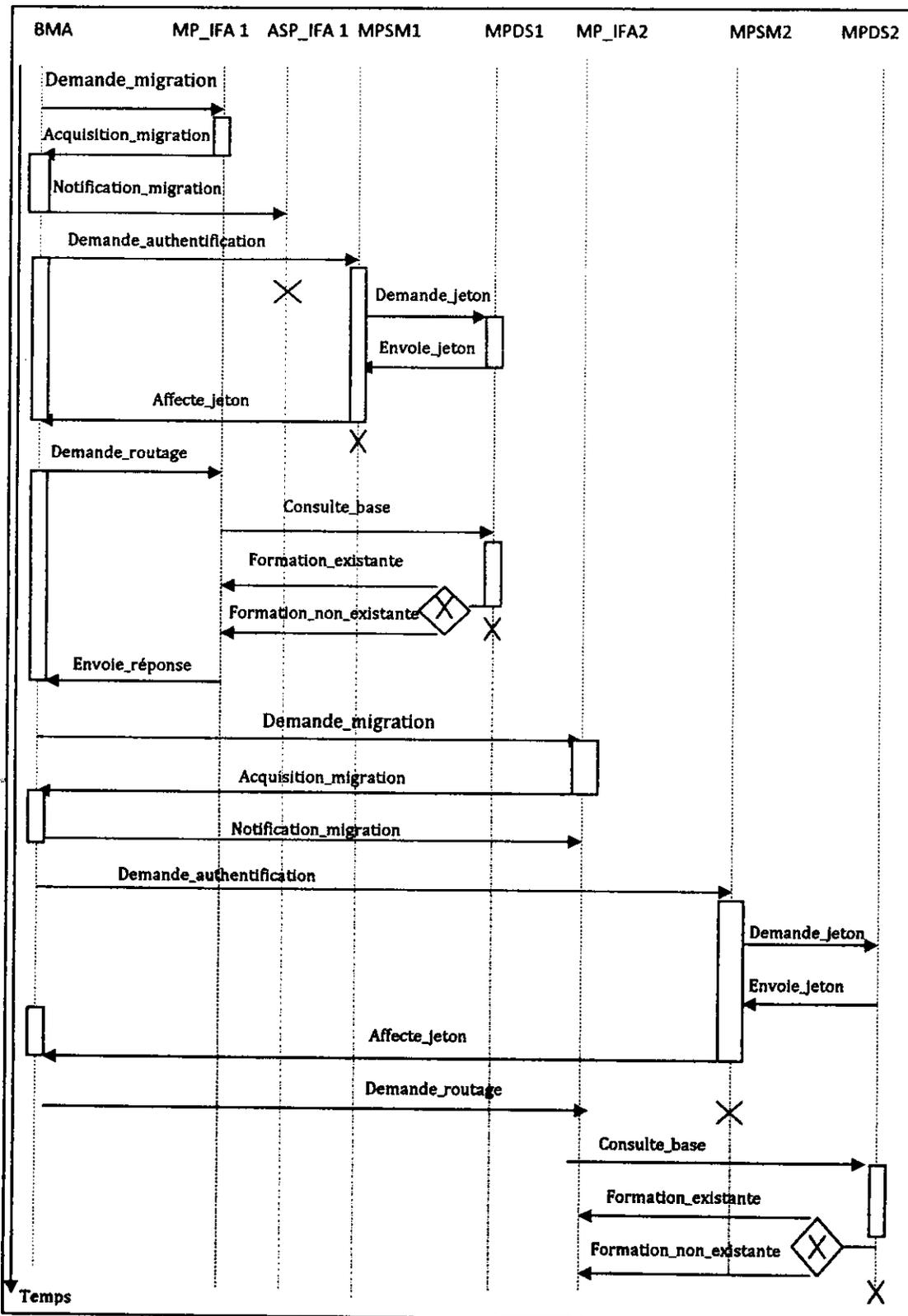


Figure IV.13 diagramme de séquence de migration de BMA vers les MPs

Scénario6 : migration de BMA vers les e-shops

- 1- BMA demande une autorisation de migration à e-shop_IFA1 de l'e-shop de son itinéraire.
- 2- E-shop_IFA1 demande à BMA d'envoyer son jeton.
- 3- BMA envoie son jeton à l'e-shop_IFA1.
- 4- e-shop_IFA1 vérifie le jeton, et accepte la demande de BMA.
- 5- BMA migre sur l'e-shop, et demande à l'e-shop_IFA de le mettre en contacte avec un ou plusieurs agents vendeurs(SMA).
- 6- L'e-shop_IFA consulte son catalogue local (LC).
- 7- Si SMA est présent sur l'e-shop, e-shop_IFA1 met BMA en contacte avec SMA.
- 8- sinon BMA va migre vers un autre site e-shop (la migration se fait dans le cas où BMA et SMA ne serions pas arrivés à un accord ou pour rechercher leur besoins dans un autre e-shop).
- 9- Tant que BMA n'a pas encore trouvé ses besoins, le processus sera répété sur tous les hôtes e-shop présents dans le site MP.

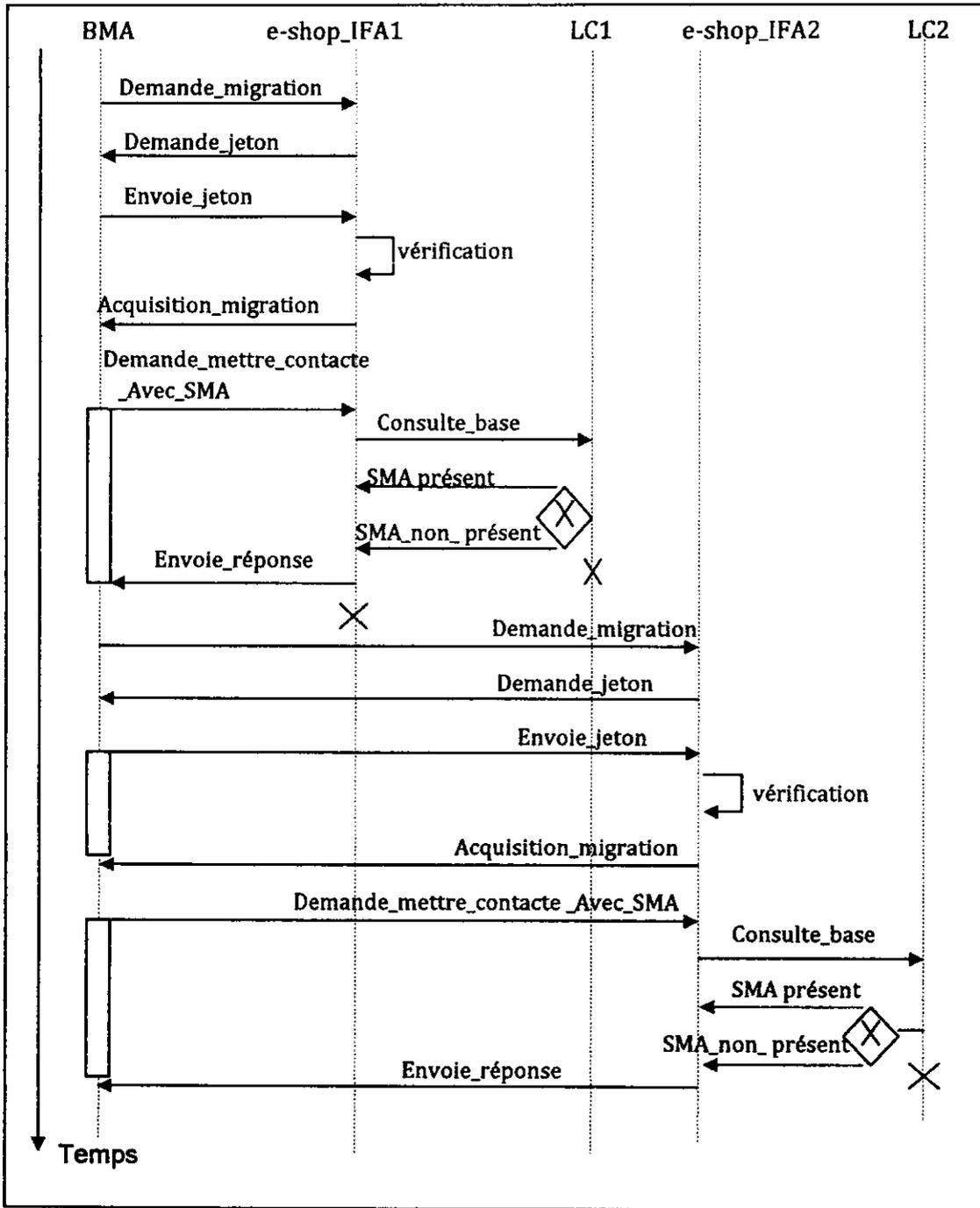


Figure IV.14 diagramme de séquence de migration de BMA vers les e-shops.

Scénario 7 : Négociation

- 1- BMA migre vers un e-shop de son itinéraire.
- 2- BMA demande à l'e-shop_IFA de le mettre en contact avec tous les SMA présents sur le l'e-shop.
- 3- e-shop_IFA vérifie sa base de connaissances si les SMA sont présents dans l'e-shop.
- 4- Si les SMA sont présents, BMA négocie avec eux selon des critères bien définis (prix, date, lieu). Jusqu'à trouver un meilleur résultat

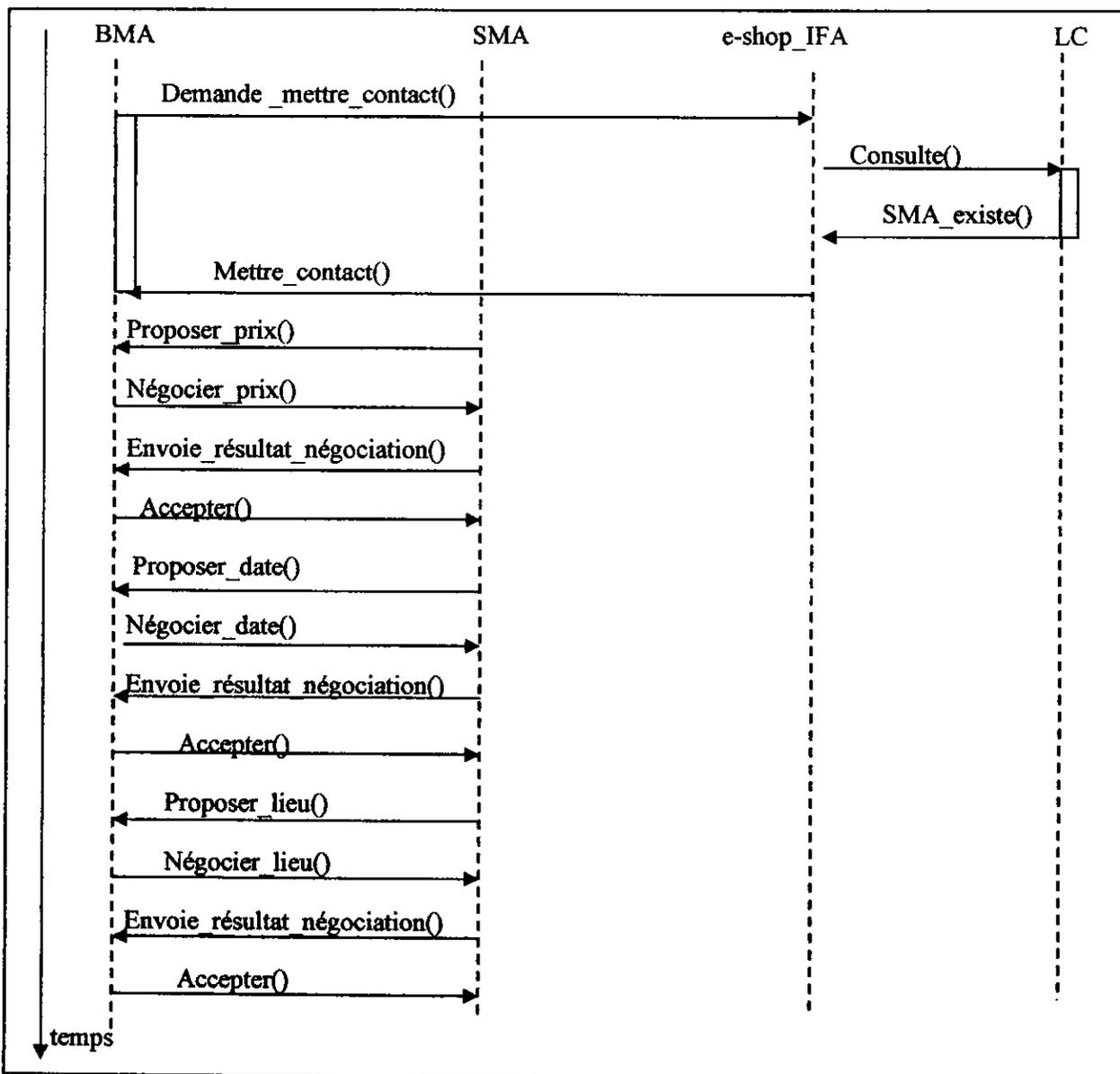


Figure IV.15 diagramme de séquence de négociation

Scénario8 : La sécurité de transaction

- 1- Au niveau d'ASP, le BMA est certifié par TSA.
- 2- Au niveau du MP, IFA_MP demande au TSA la clé publique de BMA.
- 3- TSA envoie la clé publique au MP_IFA.
- 4- MP_IFA peut décrypter les données de BMA (itinéraire, requête, signature i.e. la clé privée de BMA).
- 5- Au niveau d'e-shop, e-shop_IFA crypte les résultats par la clé publique du BMA.
- 6- Au niveau d'ASP, ASP_IFA décrypte les résultats et contrôle si la signature et l'itinéraire ont été modifiés en cours de la route.
- 7- ASP_IFA supprime le BMA.
- 8- ASP_IFA notifier la suppression de BMA au TSA.
- 9- TSA supprime le certificat de cet agent de sa base de connaissances.

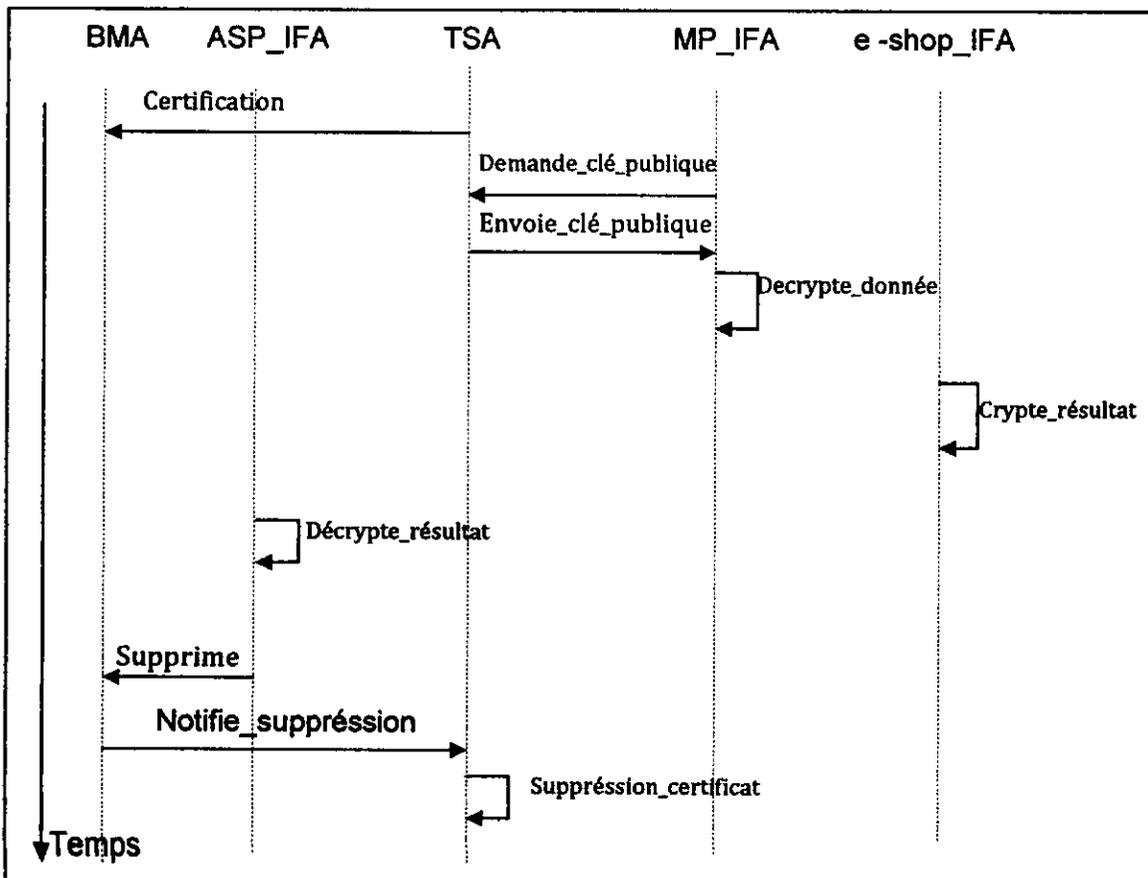


Figure IV.16 diagramme de séquence de la sécurité de transaction

Scénario 9 : Paiement

- 1- SMA demande à BMA le VAN d'acheteur.
- 2- BMA envoie ce VAN au SMA.
- 3- SMA demande à VBA de vérifier la validité de ce VAN
- 4- Si le VAN est invalide le VBA envoie au SMA un message « VAN_INVALIDE ».
- 5- Sinon le VBA envoie au SMA un message « VAN_VALIDE »
- 6- Si la réponse est VAN_VALIDE, SMA demande au VBA de vérifier le solde bancaire.
- 7- Si $\text{prix final} > \text{solde}$, SMA annule la transaction.
- 8- Sinon SMA valide la transaction.

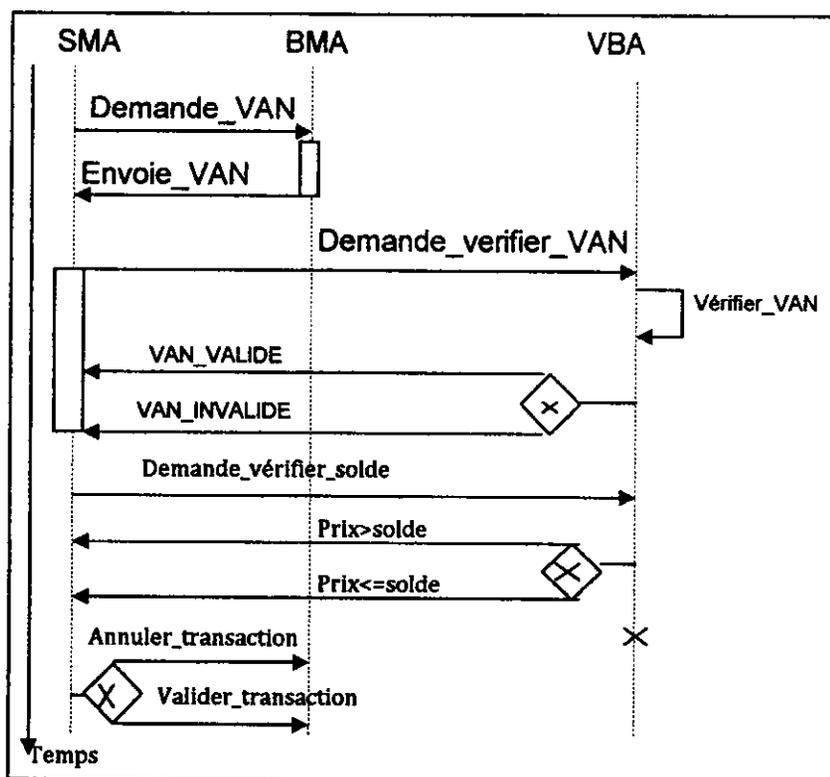


Figure IV.17 diagramme de séquence de paiement

Scénario 10: Les mises à jour de bases de connaissances

- 1- BMA effectue un achat et notifie leur validation d'achat à l'e-shop_IFA.
- 2- e-shop_IFA met à jour sa base de connaissance(LC).
- 3- e-shop_IFA demande de mettre à jour MPDS à MP_IFA.

- 4- MP_IFA met à jour leur base d'annuaire (MPDS).
- 5- MP_IFA actualise la base de MPNS ou base globale.

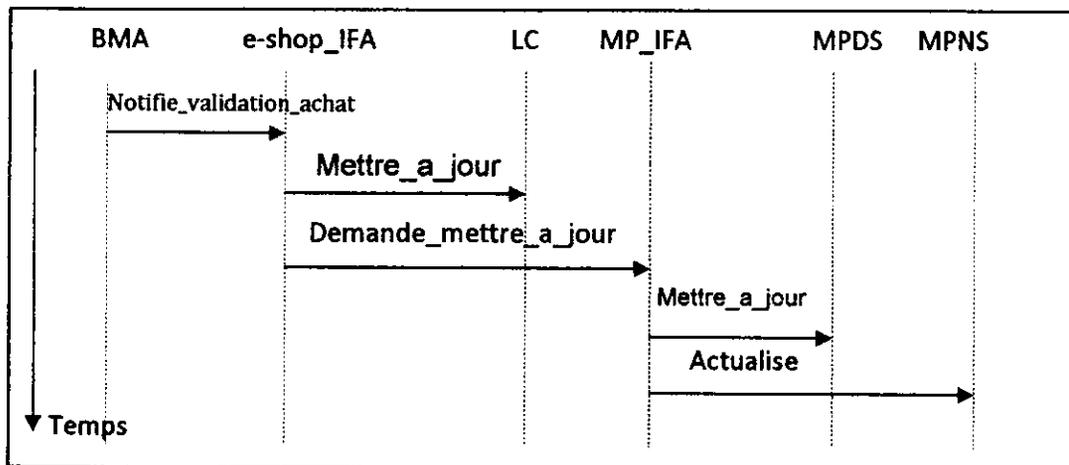


Figure IV.18 diagramme de séquence de Les mises à jour de bases de connaissances

Scénario 11: retour de BMA

- 1- Au niveau d'e-shop, BMA trouve leur besoins et demande à l'e-shop_IFA de crypter les résultats finaux.
- 2- E-shop_IFA crypte les résultats finaux par la clé publique, et l'affecte à BMA.
- 3- BMA fait un retour à l'ASP avec les résultats finaux cryptés.
- 4- ASP décrypte et contrôle ces résultats, et envoie une réponse au client.

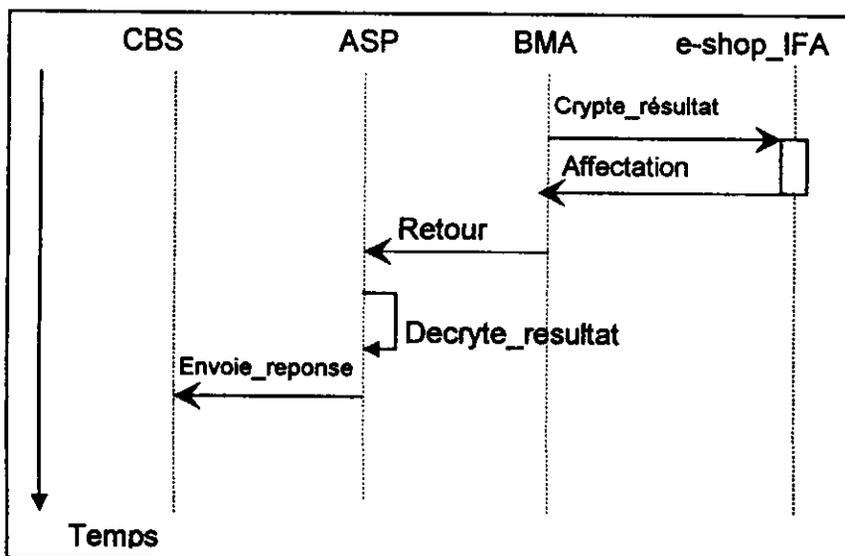


Figure IV.19 Diagramme de séquence de retour de BMA

3.2.2. Offrir une formation

Scénario1 : Identification des propositions

- 1- Un fournisseur entre le browser de site MP, et se connecte à ASP.
- 2- ASP lui envoie une applet (formulaire).
- 3- Le fournisseur remplit cette applet (formulaire) en spécifiant ses propositions (identifiant de formation, intitulé, type, domaine, date début, date fin, durée, prix min, prix max).
- 4- Le fournisseur envoie cette applet (formulaire) à l'ASP, dans ce cas le fournisseur est inscrit.

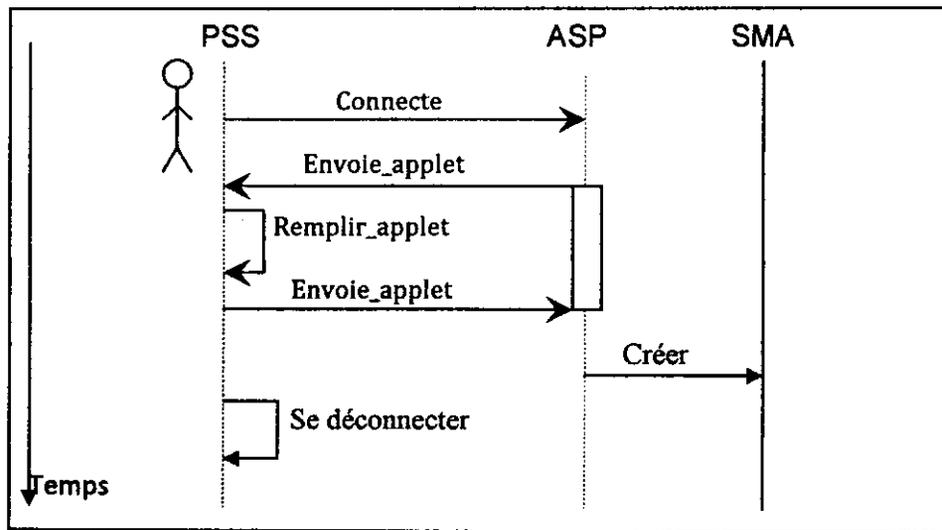


Figure IV. 20 Diagramme de séquence de l'inscription du Fournisseur

Scénario2 : création d'un SMA

- 1- Le fournisseur qui déjà spécifié ses propositions, demande à l'ASP de créer un agent mobile vendeur (SMA).
- 2- L'ASP crée le SMA en utilisant un identifiant et la requête.
- 3- L'ASP notifié le client de la création de SMA.
- 4- Dans ce cas le fournisseur est inscrit et il peut se déconnecter.

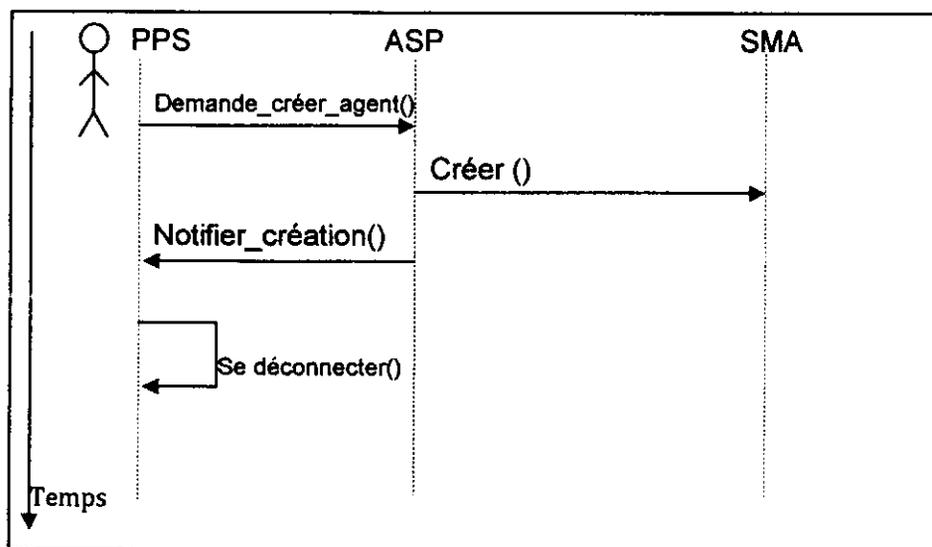


Figure IV.21 diagramme de séquence de création d'un SMA

Scénario3 : Certification de SMA

- 1- SMA demande à l'ASP_IFA de lui affecter un itinéraire initial.
- 2- ASP_IFA envoie une requête de demande d'itinéraire au MPNS.
- 3- MPNS propose un itinéraire, et il l'envoie à l'ASP_IFA.
- 4- ASP_IFA affecte l'itinéraire au SMA.
- 5- SMA demande à l'ASP_IFA d'affecter un certificat.
- 6- ASP_IFA envoie au TSA la requête et l'itinéraire, et lui demande de certifier l'agent (il crypte les données précédentes avec une clé publique et une clé privée).
- 7- TSA crée un certificat pour le SMA.
- 8- TSA envoie ce certificat à l'ASP_IFA.
- 9- ASP affecte ce certificat au SMA.
- 10-TSA sauvegarde ce certificat dans sa base de connaissances.

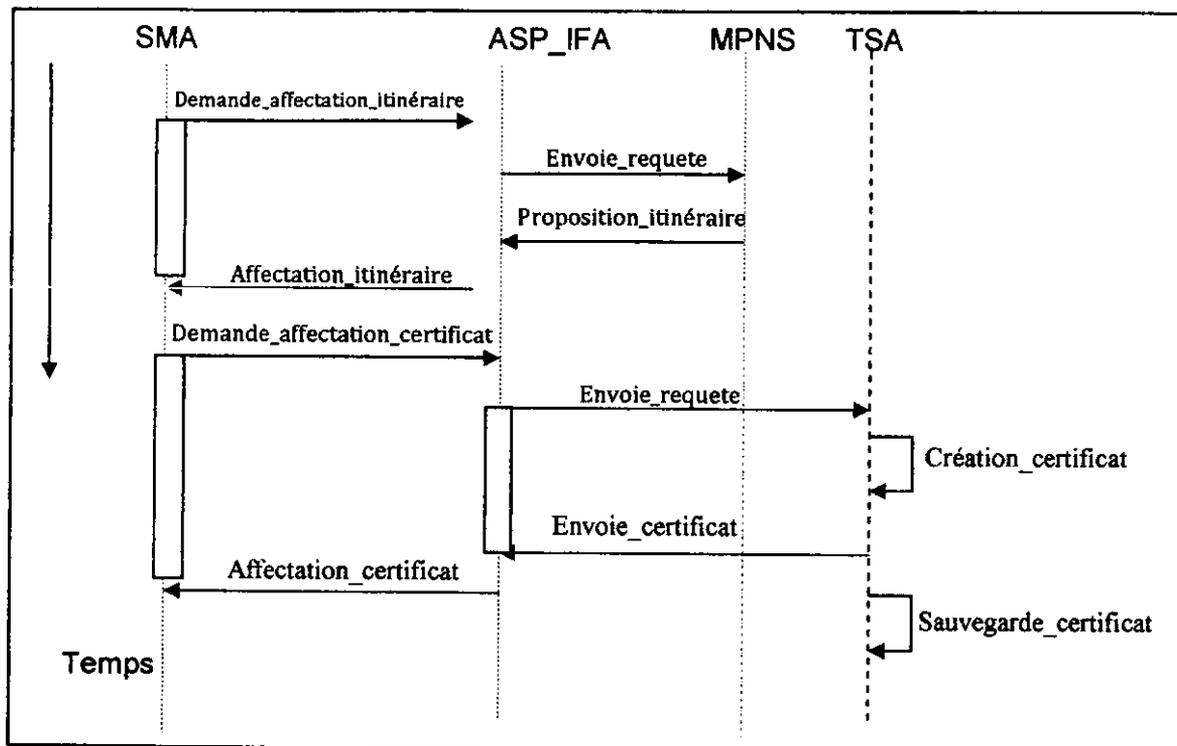


Figure IV.22 diagramme de séquence de certification de SMA.

Scénario4 : migration de SMA vers les MPs

- 1- SMA demande une autorisation de migration à MP_IFA1 du site MP1 de son itinéraire.
- 2- MP_IFA1 acquiesce la demande de SMA.
- 3- SMA notifié sa migration à ASP_IFA.
- 4- ASP_IFA initialise la durée de vie de SMA TTL (Time To Live) et le seuil de satisfiabilité.
- 5- Au niveau du site MP1, SMA demande d'être authentifié et contrôlé par le service MPSM1.
- 6- Le service MPSM1 demande un jeton au MPDS1 du site MP1.
- 7- MPDS1 envoie ce jeton au MPSM1, ce jeton permet d'accepter ou refuser le SMA.
- 8- MPSM1 affecte ce jeton au SMA.
- 9- SMA authentifié demande à MP_IFA1 de le router vers les hôtes d'e-shops.
- 10- MP_IFA1 consulte sa base (annuaire) MPDS1 pour diriger SMA vers les e-shop où se trouve BMA potentiels.
- 11- SMA effectue une vente, sinon le SMA peut migrer vers un autre site MP2 de son itinéraire, si les formations sont tous demandées ou les e-shops visités sont peu rentable i.e. le nombre de client << le nombre de formation.
- 12- Le processus de migration est répété sur tous sites MP présents dans l'itinéraire.

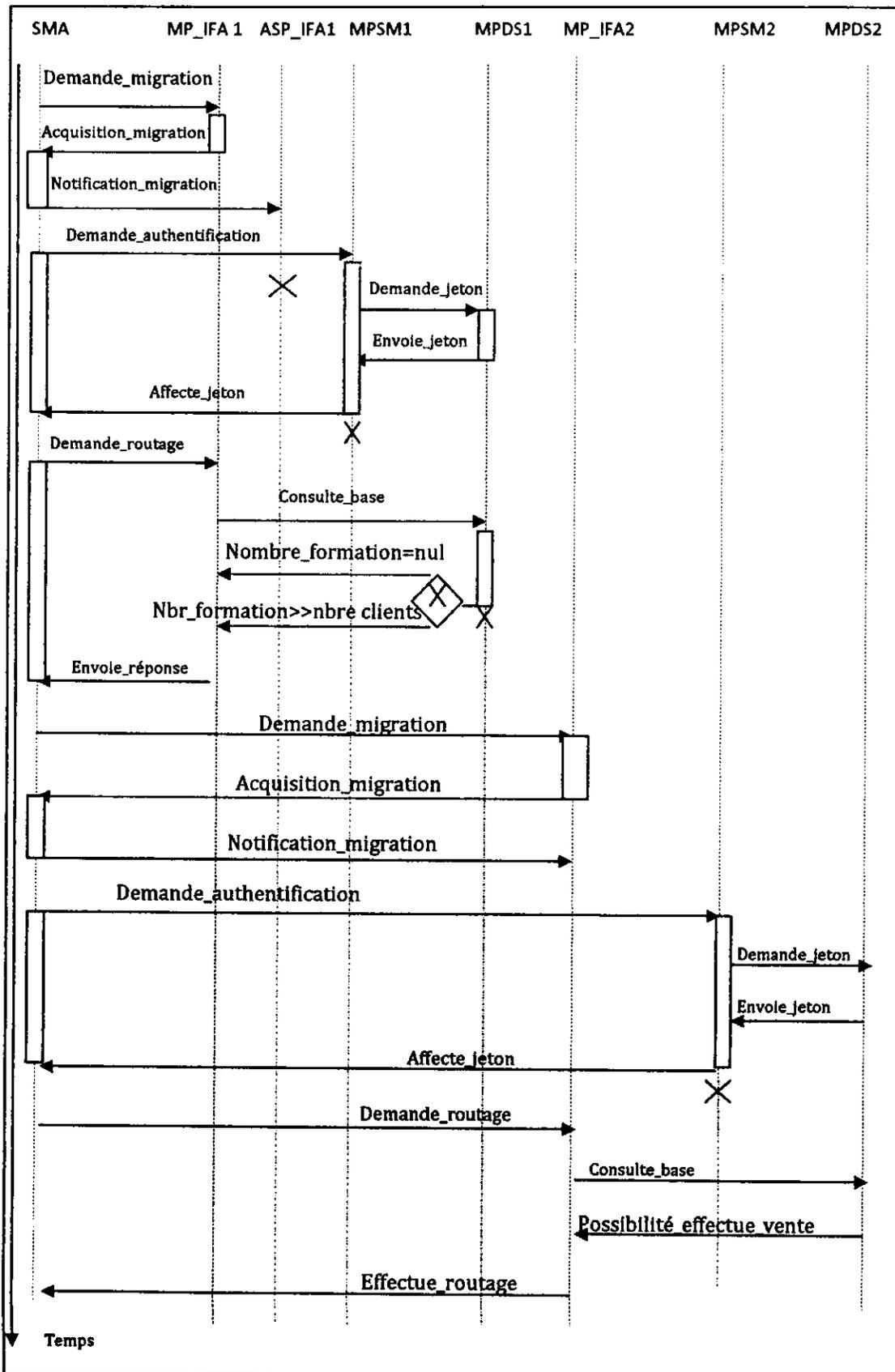


Figure IV.23 Diagramme de séquence de migration de SMA vers les MPs

Scénario5: migration de SMA vers les e-shops

- 1- SMA demande une autorisation de migration à e-shop_IFA1 de l'e-shop de son itinéraire.
- 2- E-shop_IFA1 demande à SMA d'envoyer son jeton.
- 3- SMA envoie son jeton à l'e-shop_IFA1.
- 4- e-shop_IFA1 vérifie le jeton, et accepte la demande de SMA.
- 5- SMA migre sur l'e-shop, et demande à l'e-shop_IFA1 de le mettre en contacte avec un ou plusieurs agents acheteurs(BMA).
- 6- L'e-shop_IFA1 consulte son catalogue local (LC1).
- 7- Si BMA est présent sur l'e-shop, e-shop_IFA1 met SMA en contacte avec BMA.
- 8- sinon SMA va migre vers un autre site e-shop (la migration se fait dans le cas où l'absence de BMA ou pour rechercher leur services dans un autre e-shop si TTL=0 ou nombre de formation <<nombre de clients).
- 9- Le processus est répété sur tous les hôtes e-shop présents dans le site MP.

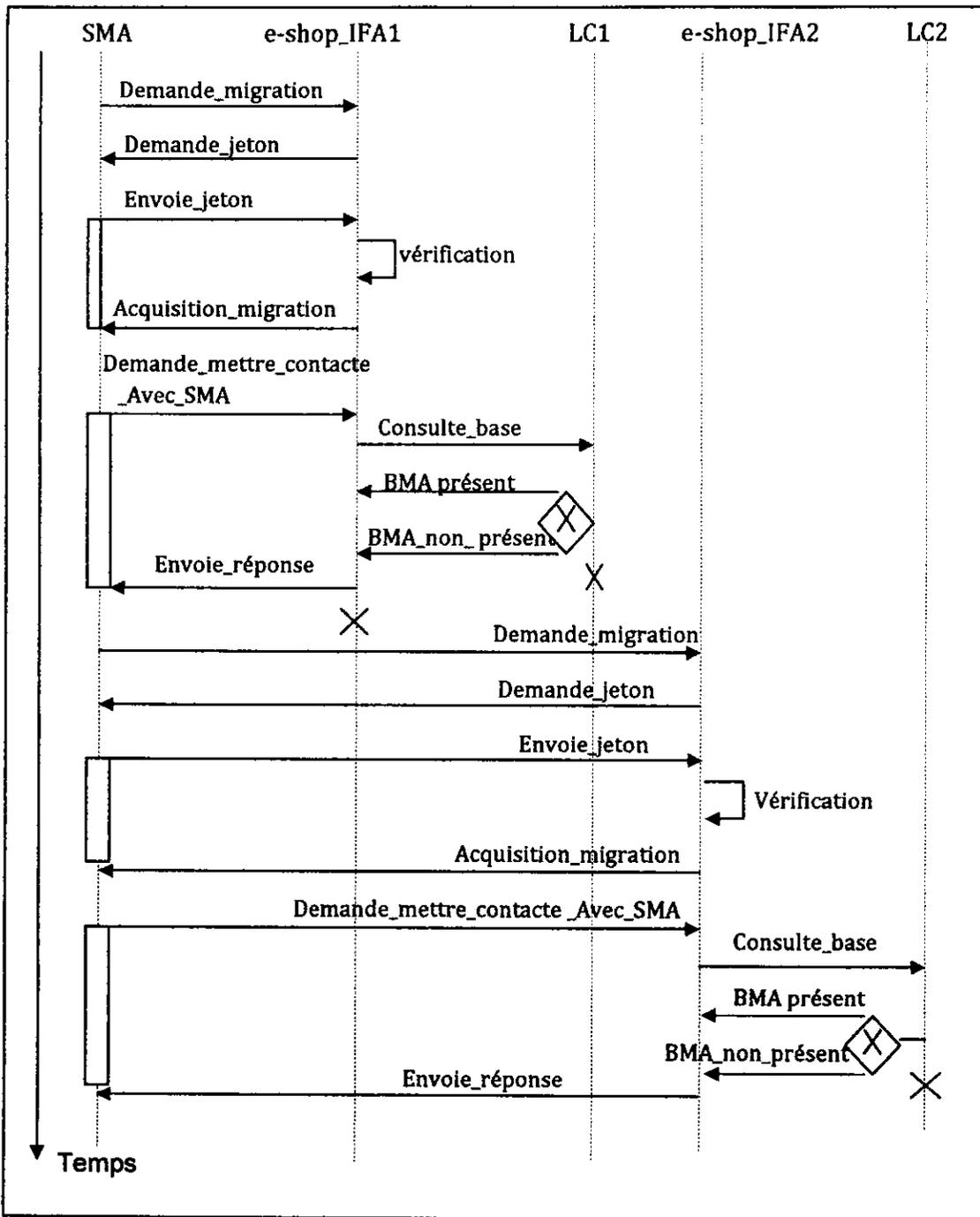


Figure IV.24 diagramme de séquence de migration de SMA vers les e-shops.

Scénario 6 : Négociation

- 1- SMA migre vers un e-shop de son itinéraire.
- 2- SMA demande à l'e-shop_IFA de le mettre en contact avec tous les BMA présents sur le l'e-shop.
- 3- e-shop_IFA vérifie sa base de connaissances si les BMA sont présents dans l'e-shop.
- 4-Si les BMA sont présents, SMA négocie avec eux selon des critères bien définis (prix, date, lieu). Jusqu'à trouver un meilleur résultat

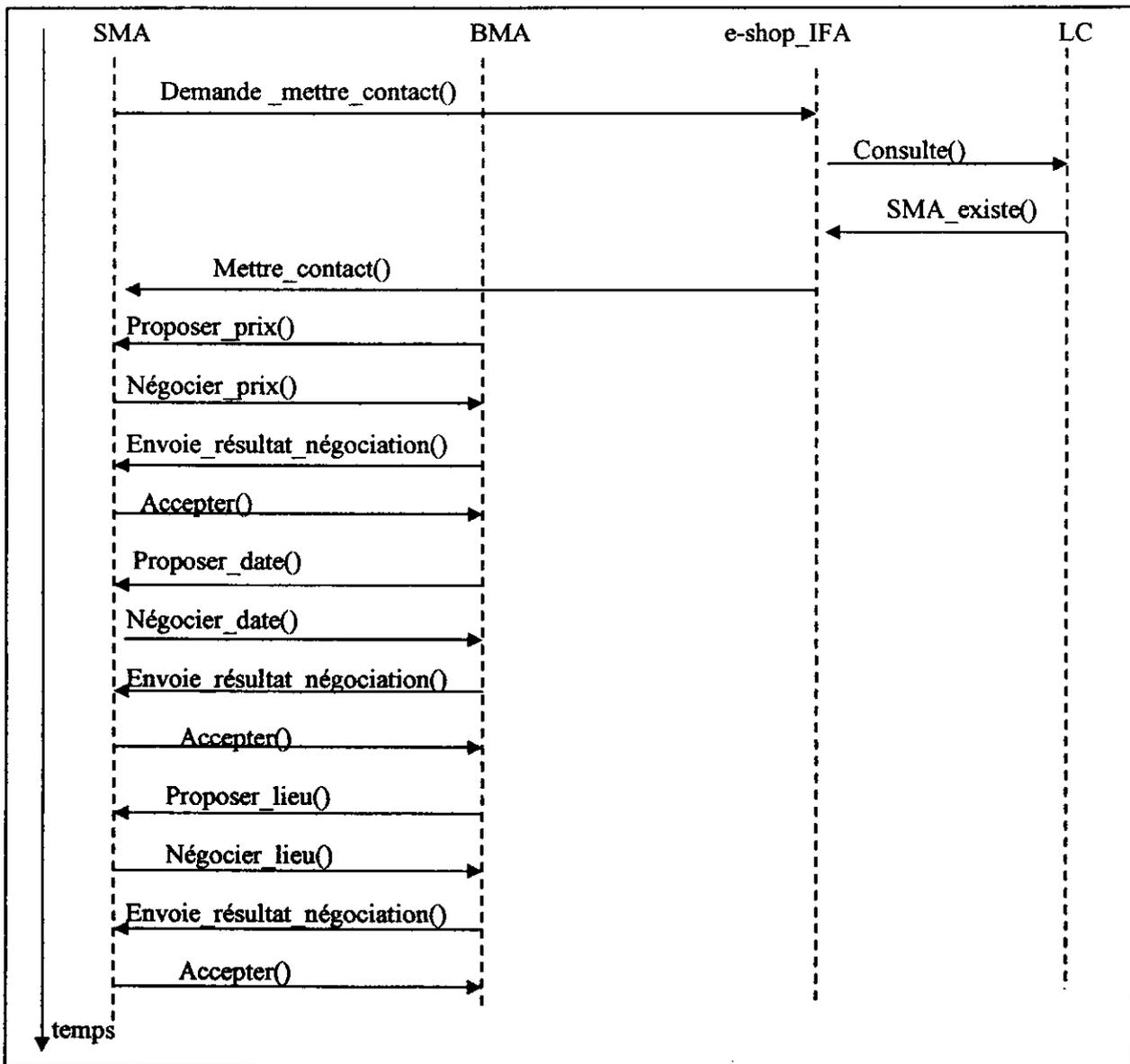


Figure IV.25 diagramme de séquence de négociation

Scénario7 : La sécurité de transaction

- 1- Au niveau d'ASP, le SMA est certifié par TSA.
- 2- Au niveau du MP, IFA_MP demande au TSA la clé publique de SMA.
- 3- TSA envoie la clé publique au MP_IFA.
- 4- MP_IFA peut décrypter les données de SMA (itinéraire, requête, signature i.e. la clé privée de SMA).
- 5- Au niveau d'e-shop, e-shop_IFA crypte les résultats par la clé publique du SMA.
- 6- Au niveau d'ASP, ASP_IFA décrypte les résultats et contrôle si la signature et l'itinéraire ont été modifiés en cours de la route.
- 7- Si SMA demande à l'ASP de renouveler son cycle de vie, dans ce cas SMA demande un nouveau certificat (TLL sera réinitialisé).
- 8- L'ASP demande au TSA de renouveler le certificat de SMA.
- 9- Sinon le SMA demande à l'ASP de supprimer son certificat.
- 10-ASP_IFA notifier la suppression de SMA au TSA.
- 11-TSA supprime le certificat de cet agent de sa base de connaissances ou sauvegarde le nouvel certificat de l'agent renouvelé.

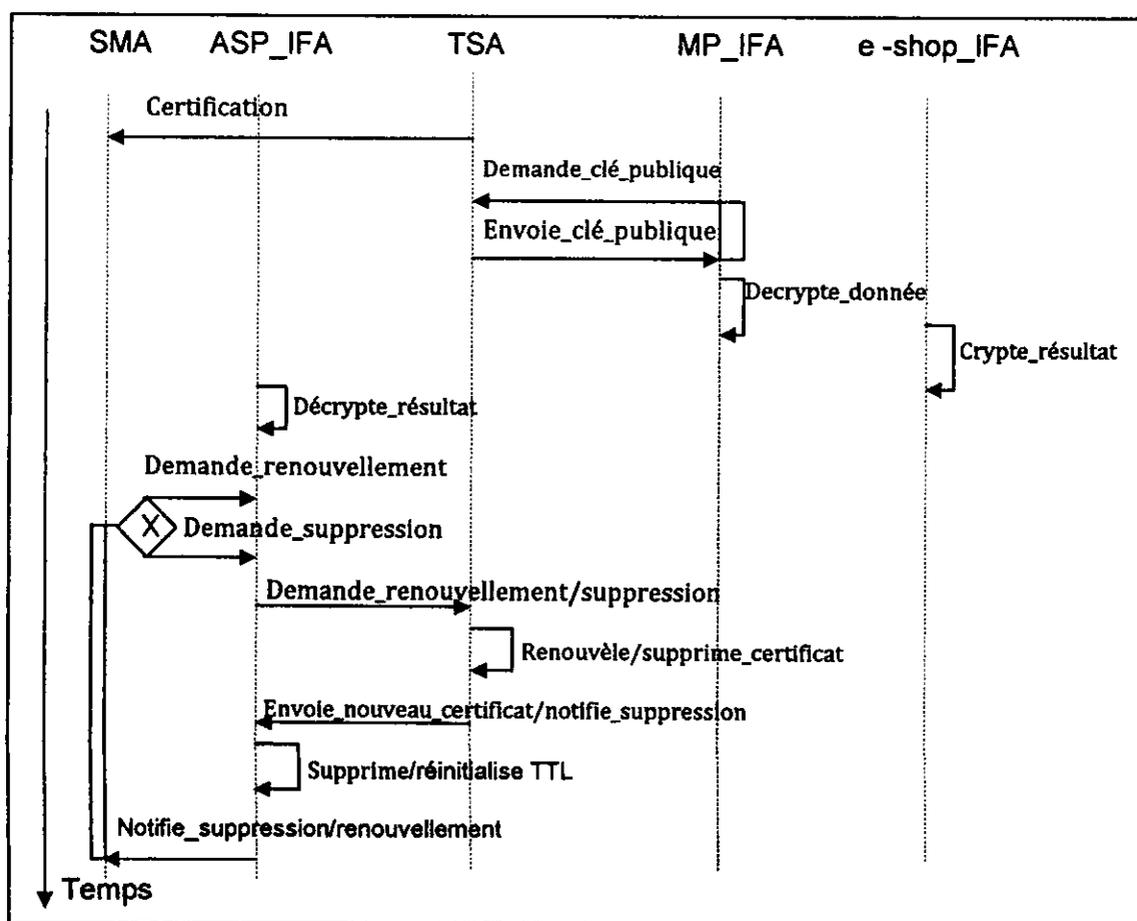


Figure IV.26 Diagramme de séquence de la sécurité d'interaction

Scénario 8: Les mises à jour de bases de connaissances

- 1- SMA effectue un achat et notifie leur validation de vente à l'e-shop_IFA.
- 2- e-shop_IFA met à jour sa base de connaissance(LC).
- 3- e-shop_IFA demande de mettre à jour MPDS à MP_IFA.
- 4- MP_IFA met à jour leur base d'annuaire (MPDS).
- 5- MP_IFA actualise la base de MPNS ou base globale.

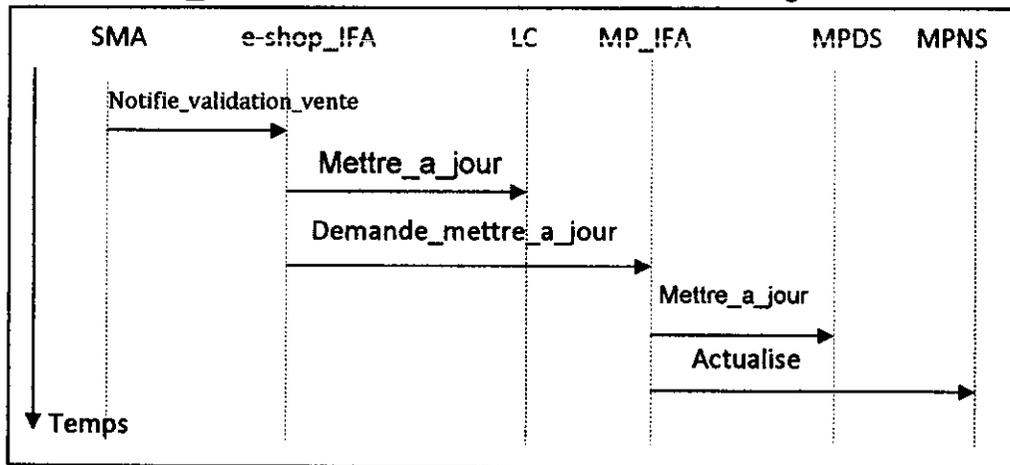


Figure IV.27 Diagramme de séquence de Les mises à jour de bases de connaissances

Scénario 10: Retour de l'agent

1. Au niveau d'e-shop, SMA effectue une vente, et demande à l'e-shop_IFA de crypter ses informations.
2. E-shop_IFA crypte ces informations par la clé publique de SMA, et l'affecte à SMA.
3. SMA fait un retour à l'ASP avec les informations cryptés.
4. ASP décrypte et contrôle ces résultats, et envoie une réponse au Fournisseur

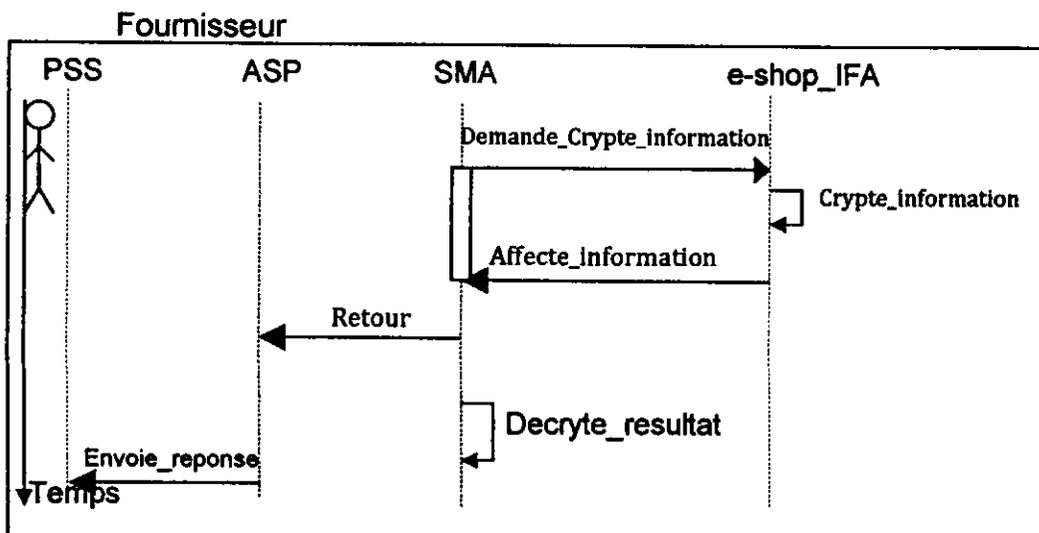


Figure IV.28 Diagramme de séquence de retour de SMA

4. CONCLUSION

Dans ce chapitre nous avons essayé d'expliquer la démarche de notre système, en utilisant quelques diagrammes d'UML pour détailler les étapes du modèle de commerce électronique pour la recherche et vente des formations sur Internet.

Chapitre V

Implémentation et Tests

1. INTRODUCTION

Dans ce chapitre nous allons présenter la mise en œuvre de notre système MPSMA décrit précédemment.

MPSMA est une place de marché virtuelle à base d'un système multiagents contenant un mélange d'agents statiques et mobiles :

✓ Les agents statiques sont principalement l'agent facilitateur de MP (MP-IFA) et un ensemble d'agents facilitateurs des e-shop (e-shop-IFA).

✓ Les agents mobiles sont les agents délégués par les utilisateurs: agents vendeurs et acheteurs créés selon les besoins.

2. CHOIX DU LANGAGE DE PROGRAMATION

Plusieurs langages destinés à la programmation d'agents, mais beaucoup de ces langages sont surtout destinés à mettre en évidence certains principes et contraintes. Java est considéré comme étant le langage le plus approprié pour la création des agents par ce qu'il dispose d'une machine virtuelle qui procure l'indépendance du système d'exploitation [MARC 03]. Nous utilisons le langage java dans cette mise en œuvre.

Il s'agit d'un langage de programmation très performant qui a été adopté par la majorité des fournisseurs. Ses caractéristiques intégrées de sécurité offrent un sentiment de confiance aux programmeurs comme aux utilisateurs des applications. De plus, Java incorpore des fonctionnalités qui facilitent grandement certaines tâches de programmation avancées comme la gestion des réseaux, la connectivité des bases de données ou le développement d'applications multitâches.

La réputation de Java en tant que langage informatique garantit que: Java est assurément un bon langage de programmation, sans doute, même l'un des meilleurs. Les caractéristiques fondamentales de Java comprennent [MARC 03] :

➤ **La distributivité** : Java possède une importante bibliothèque permettant de gérer les protocoles TCP/IP tels que HTTP et FTP. Les applications Java peuvent charger et accéder à des données sur Internet via des URL avec la même facilité qu'elles accèdent à un fichier local sur le système.

Les fonctionnalités réseau de Java sont à la fois fiables et d'utilisation aisée lorsqu'il s'agit de mettre en œuvre des tâches lourdes, comme l'ouverture d'une connexion avec un socket. De plus, Java rend plus facile l'élaboration des scripts CGI (Common Gateway Interface), et un mécanisme élégant, nommé servlet, augmente considérablement l'efficacité du traitement côté serveur, assuré par Java. De nombreux serveurs Web, parmi les plus courants, supportent les servlets. Aussi Le mécanisme d'invocation de méthode à distance (RMI) autorise la communication entre objets distribués.

➤ **La sécurité** : Java a été conçu pour être exploité dans des environnements serveur et distribués. Dans ce but, Java permet la construction de systèmes inaltérables.

➤ **La portabilité** : A la différence du C/C++, on ne trouve pas les aspects de dépendance de la mise en œuvre dans la spécification : il est possible d'utiliser le même code pour Windows 95/98/NT, UNIX, Macintosh, etc. Cette propriété est indispensable pour une programmation sur Internet.

➤ **Interprété** : L'interpréteur Java peut exécuter les bytecode directement sur n'importe quelle machine sur laquelle il a été porté. Dans la mesure où la liaison est un processus plus incrémentiel et léger, le processus de développement peut se révéler plus rapide et exploratoire.

➤ **Multithread** : Les avantages du multithread sont une meilleure interactivité et un meilleur comportement en temps réel.

➤ **Sérialisation** : Java fourni un mécanisme de sérialisation d'objets permettant d'échanger les instances entre JVM différentes. Ce mécanisme permet de traduire un graphe d'objet Java en un flot d'octets qui peut être redirigé vers un fichier ou vers une autre machine à travers le réseau. Pour qu'une instance soit sérialisable, il faut que sa classe implémente l'interface *Sérialisable*. La sérialisation d'une instance revient à écrire tous ses champs dans un flot d'octets. Lorsque l'un de ces champs est une référence à un autre objet, l'objet référencé est aussi sérialisé. Afin de spécialiser le processus de sérialisation, Java permet de définir dans une classe la méthode *writeobjet()* qui surcharge le mécanisme de sérialisation pour les instances de cette classe. La dé-sérialisation est le processus inverse, de la sérialisation. Le graphe d'objet peut être reconstruit à partir du flot d'octets obtenu par la sérialisation. La méthode permettant la surcharge est *readObjet()*.

3. RÉALISATION DU SYSTÈME MPSMA

3.1. Implémentation

En termes d'architecture, MPSMA comprend :

- Une ou plusieurs place de marché virtuelle MP.
- Un environnement d'exécution ASP.
- Une autorité de sécurité TSA.
- Une autorité bancaire virtuelle VBA.

Pour assurer le bon fonctionnement de notre système MPSMA, nous l'avons associé à un environnement d'exécution d'agents destiné à supporter les codes des agents statiques et mobiles contenant MPSMA, en les permettant :

- 1- Leurs créations.
- 2- Leurs mobilités.
- 3- Leurs communications.
- 4- Leur sécurité.
- 5- leurs exécutions.

Nous allons détailler maintenant ces cinq points :

3.1.1. Implémentation de la création des agents

La classe agent est une class java, à partir de laquelle tous les autres agents seront hérités par la suite. La figure ci-dessous illustre la forme de notre classe agent :

```
public class agent implements Runnable {
public String adrIP;
public Requet requet;
private Code code;
public agent() {...}
void initialise() {...}
void set_code(Code cd) {...}
Requet get_code(Code cd) {...}
public void run() {...};
public void stop() {...};
...
}
```

Figure V.1 La forme de la class agent.

Chaque agent est représenté par un **identifiant** (classe Code). Cet identifiant contient toutes les informations permettant de retrouver un agent. C'est à dire l'adresse de la machine qui fait tourner l'infrastructure et la clé de celle-ci. Il contient aussi le nom de l'agent. Ce nom lui est donné par l'ASP lors de l'enregistrement. L'identifiant est susceptible d'être utilisé comme clé dans une table de hachage et d'être comparé souvent. Pour accroître la performance des comparaisons, nous avons surchargé la méthode `hashCode()` et `equals()`. En java, la fonction `hashCode()` renvoie la clé de hachage de l'objet sur un entier.

3.1.2. Implémentation de la mobilité des agents

Le mécanisme de sérialisation de java, nous permet de transformer l'ensemble des objets gérés par l'agent (un graphe d'objet) en un flot d'octets, de déplacer ces données vers le site destination comme étant un message, puis reconstruire le contexte. Le déplacement d'un agent suit les étapes suivantes :

1. Sérialisation du contexte de l'agent et production d'un message incluant le contexte sérialisé de l'agent et son code.

2. Envoie du contexte et du code de l'agent vers le site destination.
3. Destruction de l'agent sur le site origine.
4. Réception du message sur le site destination.
5. Création d'un chargeur pour cet agent (le mécanisme de chargeur de classe de java nous permet de transformer le code de l'agent reçu dans le message en une classe java sur le site destination).
6. Dé-sérialisation du contexte de l'agent.
7. Démarrage de l'agent sur le nouveau site.

La classe `agentmobile.java`, implémente l'interface `java.io.serializable` pour autoriser la migration de son code.

L'identificateur d'un agent mobile est un objet *id* de type `Code`. C'est la référence de l'agent ou la correspondance entre un serveur visité et l'*id* de l'agent sur ce serveur. Les méthodes `getLocalID` et `getAllServers` de la classe `Code` renvoient respectivement l'*id* de l'agent sur le serveur courant et l'ensemble des serveurs visités par l'agent.

Pour assurer la **tolérance de panne**, il est nécessaire de ne pas faire correspondre via le champ *id* un *id* local à un objet de type `agentmobile`. En effet, la référence à un agent mobile n'est valide que durant la vie de cet agent. S'il est détruit, et qu'un nouveau agent est créé la référence est invalide.

Aussi, nous utilisons comme référence à un serveur, son adresse au sens RMI. C'est une chaîne de caractère de type URL : `http://machine:port/nom`.

Voici la représentation simplifiée de la forme de la classe agent mobile :

```
public class agentmobile extends agent implements java.io.Serializable{
    private String originalServerAddress;
    private String fromServerAddress;
    private String currentServerAddress;
    agentmobile(){super();
    ...}
    public FileOutputStream seriealise(){
    ...}
    public agentmobile deserealise(FileOutputStream A1){
    ...}
    void mouve(String machine,int port) throws UnknownHostException,
    IOException{...}
    ...
}
```

Figure V.2 La forme de la class agent mobile.

3.1.3. Implémentation de la communication inter-agents

La communication inter-agents que nous avons adoptés est basée sur le langage KQML. Pour cela notre interface java `message` définit un message avec les éléments KQML :

- ✓ **Sender** : l'expéditeur du message.
- ✓ **Reciver** : le destinateur du message.
- ✓ **ContentType** : le type du contenu. Il peut être :
 - **QueryMessage** : lorsque le message est une requête.
 - **ReponseMessage** : lorsque le message est une réponse.
 - **InformativeMessage** : pour les messages d'informations.

La figure ci-dessous illustre les interfaces de communications inter-agents :

```
package packageMP;

public interface message {
    .....
}

public interface messageacheteur extends message {...}

public interface messagevendeur extends message {...}

public interface InformativeMessage extends message {...}
```

Figure V.3 forme de l'interface message.

Les agents qui veulent **communiquer** entre eux doivent implémenter l'interface message. Quand un agent implémente cette interface, il est capable de recevoir des messages en provenance d'un canal de communication.

3.1.4. Implémentation de la sécurité des agents

La sécurisation est garantie par le système MPSMA

Tout utilisateur inscrit dans MPSMA sera certifié à l'aide d'une autorité de sécurité TSA. Aussi, ses données seront cryptées en se basant sur l'algorithme RSA. Pour cette raison nous avons construit une classe java **MyRSA.java** et la classe **certificat.java**.

3.2. Implémentation de l'exécution des agents

3.2.1 Implémentation de l'agent acheteur

Un agent acheteur est une instance java étant la class **agentmobile.java** illustré précédemment, il représente un acheteur réel intéressé par une formation en ligne, qui peut exister probablement sur le marché électronique. L'agent acheteur est contrôlé par l'utilisateur qui le dote d'un ensemble d'informations concernant le type de formation, la date, et le prix d'achat. Un agent acheteur dans MPSMA est capable de communiquer avec l'agent facilitateur du MP afin de ramener les informations et les descriptions de prix de toutes les formations d'un type bien déterminé.

Pour bien accomplir sa tâche, chaque agent acheteur doit disposer l'ensemble des informations suivantes :

- ✓ l'identifiant de l'acheteur associé.
- ✓ la description de la formation recherchée.

La classe **agentacheteur.java** hérite la class **agentmobile.java**. La forme de notre classe agent acheteur est représentée dans la figure suivante :

```

public class agentacheteur extends agentmobile implements
messageacheteur{

    msg msg;
    Client client=null;
    Formation formation=null;
    Certificat certificat=null;

    public agentacheteur() {..... }
    public void close() {....}
    public Formation getformation() {.... }
    public void setformation(Formation formation) {....}
    public Client getclient() {.... }
    public void setclient(Client client) {....}
    public msg getmsg() {...}
    public void receive(agentvendeur V,msg msg){....}
    public void receive(agentfacilitateur F,msg msg){...}
    public void setmsg(msg msg) {... }
    public msg getmsg(msg msg) {.... }
    public Client getselerdetails(String id) {.... }
    boolean compare(Formation f1,Formation f2){...}
    public String négotiate(agentacheteur A,agentvendeur V){...}
    public void afternegociate(String result){}
    public void setSellerSender(agentacheteur A) {...}
    public void setByerReciver(agentvendeur V) {...}
    public void initalise(){...}

    public void sendMessage(agentacheteur A,agentvendeur B, message
msg ){...};

    public void sendMessage(agentacheteur A,agentfacilitateur F,
message msg ){...};

    ...}
}

```

Figure V.4 forme de la class agent acheteur.

3.2.2. Implémentation de l'agent vendeur

Un agent vendeur, représente un vendeur réel désire la vente de ses formations aux acheteurs qui se trouvent probablement connectés sur MPSMA. L'agent vendeur

expose ses formations aux clients qui sont des agents acheteurs, peut négocier et effectuer des transactions.

Un agent vendeur publie, dès son inscription, ses offres de vente. Les informations de l'agent vendeur comprennent :

- ✓ l'identifiant du vendeur.
- ✓ l'adresse du vendeur.
- ✓ la liste des formations à vendre.

La classe **agentvendeur.java** hérite de la class **agentmobile.java**. La forme de notre classe agent vendeur est illustrée dans la figure suivante :

```

package packageMP;

public class agentvendeur extends agentmobile implements
messagevendeur{

    msg msg;
    Client client=null;
    Formation formation=null;
    Certificat certificat=null;
    long ttl= 365;

    public void setTimeToLive(long ttl ) {..... }
    public long getTimeToLive() {..... }

    public agentvendeur() {..... }
    public void close() {....}
    public Formation getformation() {.... }
    public void setformation(Formation formation) {....}
    public Client getclient() {.... }
    public void setclient(Client client) {....}
    public msg getmsg() {...}
    public void receive(agentacheteur A,msg msg){....}
    public void receive(agentfacilitateur F,msg msg){...}
    public void setmsg(msg msg) {... }
    public msg getmsg(msg msg) {.... }
    public Client getbyerdetails(String id) {....}
    boolean compare(Formation f1,Formation f2){...}
    public String négotiate(agentacheteur A,agentvendeur V){...}
    public void afternegociate(String result){}
    public void setByerSender(agentvendeur A) { ...}
    public void setSellerReciver(agentacheteur V) {...}
    public void initalise(){...} ;
        public void run(){...};
        public void stop(){...};

    public void sendMessage(agentvendeur V,agentachteur A,
message msg ){...};

    public void sendMessage(agentvendeur V,agentfacilitateur F,
message msg ){...};

    ...}

```

Figure V.5 forme de la class agent vendeur.

3.2.3. Implémentation de l'agent facilitateur du MP

La médiation est le processus d'association des acheteurs à des vendeurs dans le but d'aboutir à une entente. Cette étape peut être automatisée à l'aide des agents facilitateurs. Ces agents sont capables de trouver les offres les plus intéressantes en consultant leurs bases d'informations et les communiquant aux agents acheteurs.

L'agent facilitateur du MP gère une **MPDS** contenant des identifications sur tous les vendeurs inscrits auprès de la MP. Pour cet effet, l'**MP_IFA** peut :

- répondre à l'acheteur qui demande des informations sur les différentes formations existantes.
- Trouver les vendeurs qui correspondent à la demande de l'acheteur.
- Informer les acheteurs de l'existence de nouvelles formations.
- Gère les interactions et les négociations entre les différents agents acheteurs et vendeurs.

En tout moment, l'agent facilitateur peut être associé à plusieurs transactions entre différents acheteurs et vendeurs. Il doit garder la trace des transactions en mémorisant les informations nécessaires afin de pouvoir suivre leurs évolutions dans le temps et gérer les interactions jusqu'à leur achèvement.

Les informations concernant l'agent facilitateur sont :

- la liste de tous les agents acheteurs inscrits auprès de la MP gérée par cet agent facilitateur.
- La liste de tous les vendeurs inscrits dans la MP gérée par ce facilitateur.
- La liste de toutes les formations disponibles actuellement dans sa MP.

Un agent facilitateur est un agent **statique** hérité de la classe **agent.java**. La figure suivante illustre la forme de la classe **agentfacilitateur.java** :

```
public class agentfacilitateur extends agent implements
messagefacilitateur {

    MyRSA RSA;

    msg msg;

    private agentvendeur vendeur;

    private agentacheteur acheteur ;

    private java.sql.Connection connection;

    ResultSet resultat;

agentfacilitateur() throws Exception{
}

public void sendMessage(agentacheteur A, msg msg ){...};
public void sendMessage(agentvendeur V, msg msg ){...};
void addAgentvendeur(agentvendeur vendeur){...};
void addAgentacheteur(agentacheteur acheteur){...};
public void finishSendingRequest() {...}
public void initializeMessage() {...}
public void achat(agentacheteur A,agentvendeur V) {...}
public msg getmsg() {...}
public void receive(agentacheteur A,msg msg){...}
public void receive(agentvendeur V,msg msg) {...}
public void setmsg(msg msg) {...}
public msg getmsg(msg msg) {...}
boolean iscomptible(Formation f1,Formation f2) {...}
public void verifiercertificat(String publickey){... }
    public void ConnectBD(){... };
    public void initalise() {...};
    public void actualiseMPDS() {...};
}
```

Figure V.6 La forme de la class agent facilitateur.

4. RÉALISATION DE L'APPLICATION

Notre application contient deux sites :

1- Un site utilisateur : permet l'inscription des clients ainsi la spécification des leurs besoins qui peuvent être des formations informatiques à commander ou à vendre en ligne. L'inscription d'un client lui permet de déléguer un agent mobile acheteur (ou vendeur) afin d'effectuer les transactions commerciales à sa place.

2-un site MP (ou Marquet Place) virtuel : géré par un ensemble d'agents statiques, les agents facilitateurs de la place marché. La MP assure :

- La commande des formations informatiques en ligne.
- L'offre des formations informatiques en ligne.

4.1. Réalisation de la commande d'une formation en ligne

Dès qu'un client démarre notre système, il rencontre un catalogue en ligne sous forme d'une page web publicitaire sur des formations informatiques (figure V.1) chacune joue le rôle d'une racine ou bien représente une famille de même type, indexée à une autre page mieux détaillée sur la formation elle-même (exemple : figure V.2) afin de conduire le client à bien définir ses exigences : c'est **La spécification des besoins**.

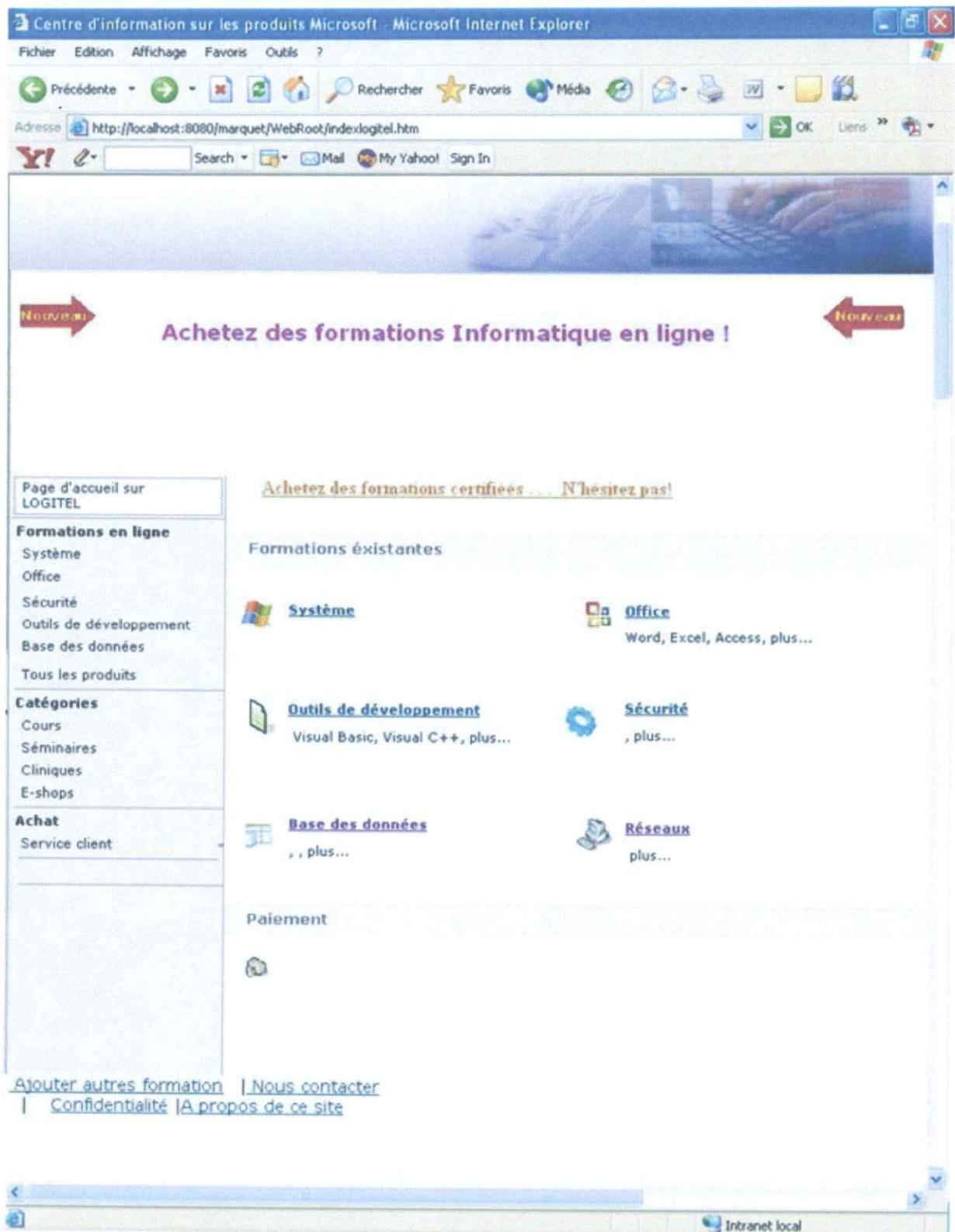


Figure V.7 Index du site.

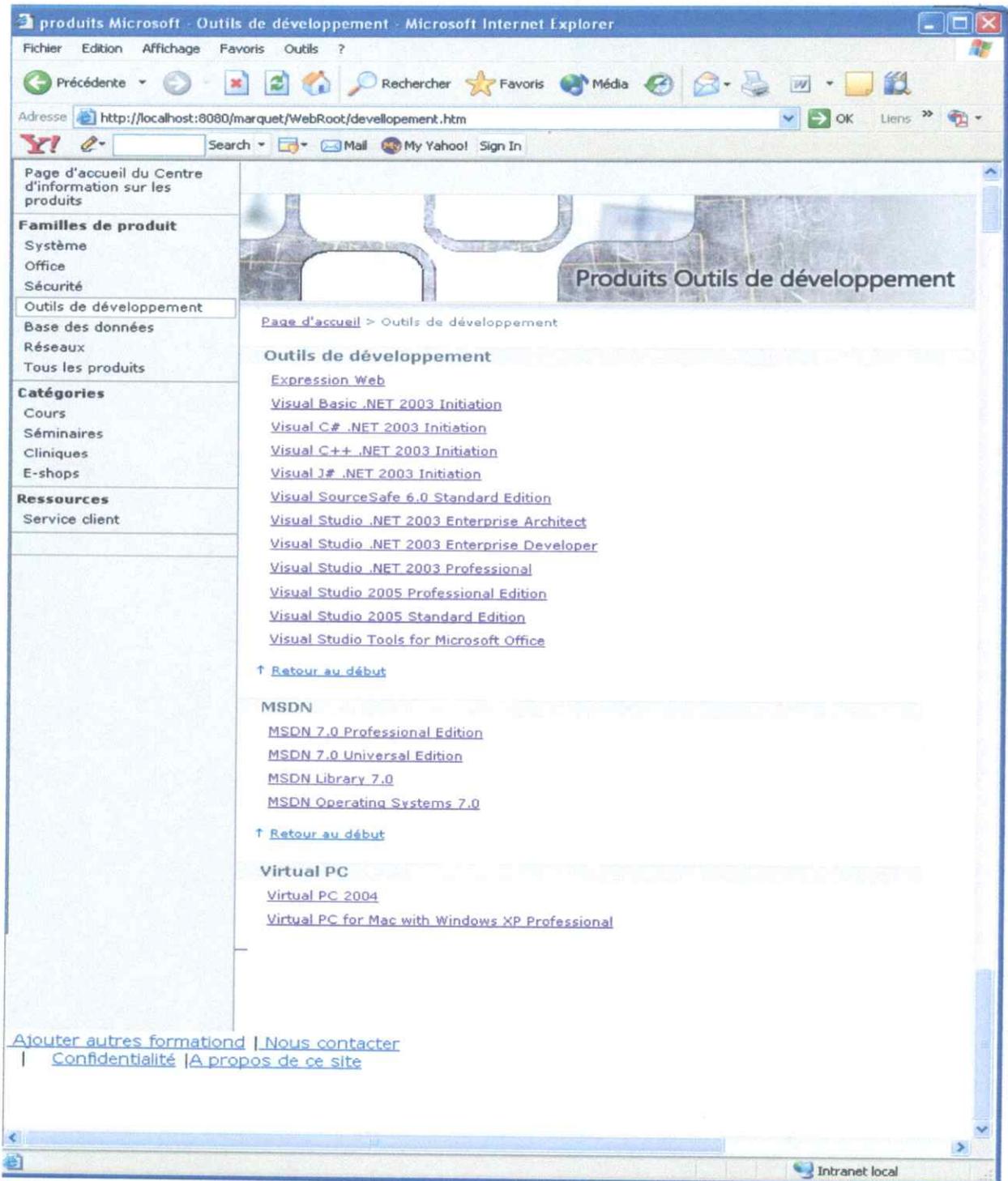


Figure V.8 Exemple de description d'une formation sélectionnée.

Après cette étape, le client trouvera une fenêtre gérée par l'ASP contenant un formulaire d'inscription (figure V.3), en remplissant ce formulaire, le client sera invité pour la validation de son choix dans la page suivante.

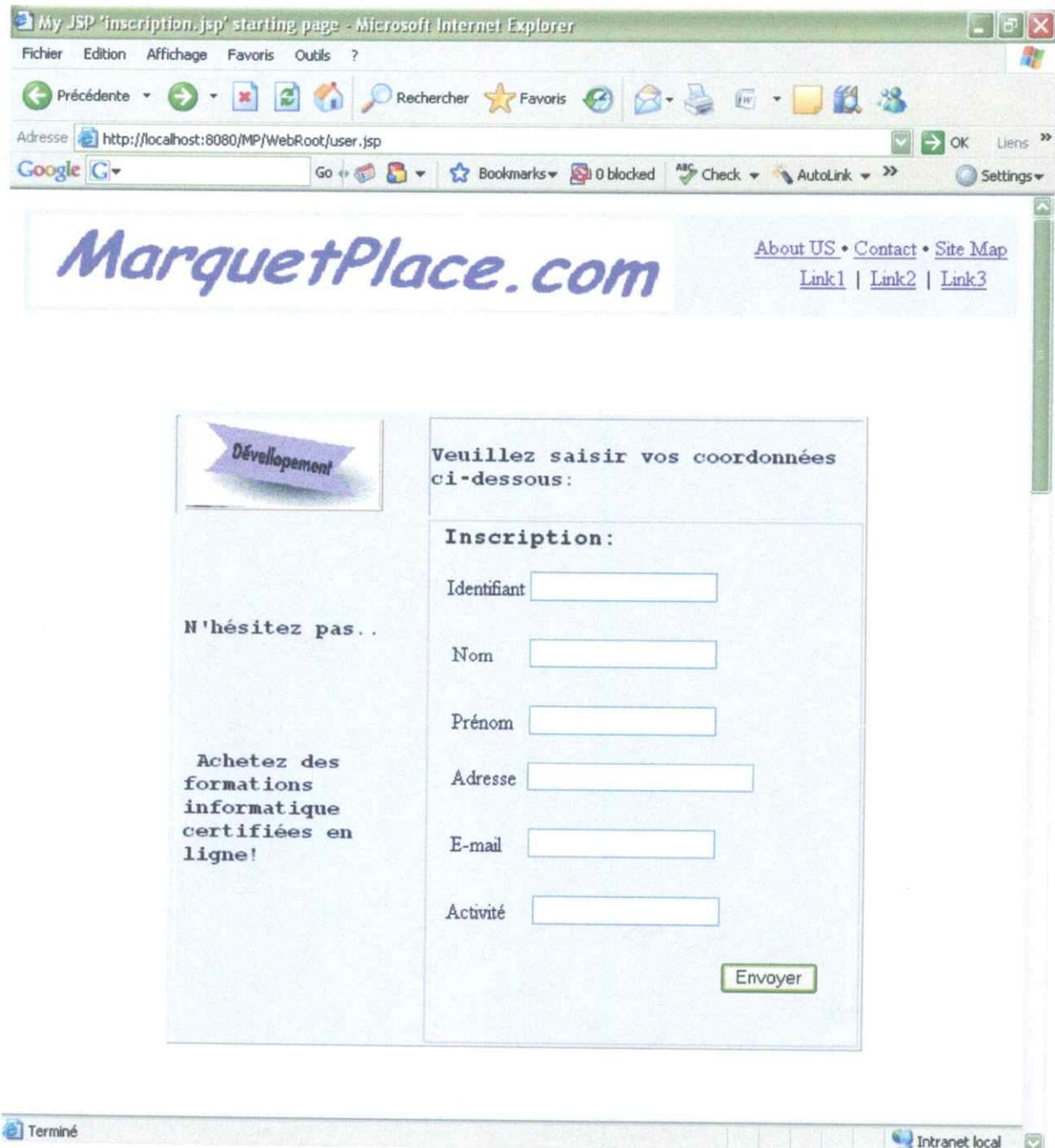


Figure V.9: formulaire d'inscription

La figure ci-dessous (figure V.4) représente la validation des besoins du client ainsi la création de son agent délégué. En terminant ces deux tâches, ce client soit bien inscrit dans une base de données que seul l'agent facilitateur peut y accéder, un agent mobile lui représente dans la place de marché.

The screenshot shows a Microsoft Internet Explorer browser window displaying a web form. The browser's address bar contains the URL: `Client.jsp?identifiant=mypass&nom=nserzr&prezr=nserzr&adr=&email=&tel=&activite=&statu=&ok=Envoyer`. The form is titled "l'agent délégué:" and contains the following elements:

- Identifiant:** A text input field.
- Type:** A dropdown menu with "Agent acheteur" selected.
- Formation à acheter :** A section header followed by several sub-fields:
 - Domaine:** A text input field.
 - Description:** A dropdown menu.
 - Date désiré:** A text input field.
 - Prix min:** A text input field.
 - Prix max:** A text input field.
 - type:** A dropdown menu.
- Envoyer:** A button to submit the form.

The browser's status bar at the bottom shows "Terminé" and "Intranet local".

Figure V.10 : spécification des besoins d'un client

Dès l'enregistrement du client et son choix, son agent acheteur va suivre le processus commercial à sa place en toute sécurité en migrant vers l'agent facilitateur du MP suivant l'itinéraire affecté par le MPNS. L'agent facilitateur va traiter sa requête et lui diriger vers l'e-shop adéquate en consultant sa base de connaissance locale MPDS (communication inter-agents utilisant KQML).

A la fin, l'agent mobile acheteur reviendra à son client lui ramenant un rapport commercial sur toutes les formations trouvées après la négociation du prix, de la date et de lieu. Le client choisit une et affirmera son OK et passera à l'étape de paiement. Dans cette étape nous avons construit un bon de commande contient les informations de client (Nom, Adresse, etc.), les informations de formation commandée (prix d'achat, lieu, domaine, intitulé, date début, date fin, durée) et les informations concernant le fournisseur (l'adresse, le nom, etc.).

Le bon de commande sera rempli automatiquement selon le résultat de la transaction commerciale et envoyé sous forme d'un e-mail du client ou fournisseur.

http://127.0.0.1:8080/marquet/WebRoot/boncommande.jsp - Vista Code Name:Invincible

Echier Edition Affichage Favoris Outils ?

Précédente Recherche Favoris OK Links Ouvrir

Adresse http://127.0.0.1:8080/marquet/WebRoot/boncommande.jsp

Rechercher sur le Web

Google Envoyer Mes favoris 36 bloqué(s) Orthographe Traduire Envoyer à Paramètres

EPSON Web-To-Page Imprimer Aperçu avant impression

A:

De:

Objet:

Bon de commande le

Client:
nom:
prenom:
adresse:

Fournisseur:
nom:
telephone:
adresse:

Achat d'une formation:
domaine :
pour jours
entre
et
prix de à

Pour confirmer votre commande, cliquez sur le bouton Confirmer et envoyer.

Envoi Message

Le Dictionnaire Google est téléchargé en arrière-plan Internet

Figure V.11 : Bon de commande

Les deux figures ci-bas montrent la phase de spécification des besoins du client en question :

The screenshot shows a Microsoft Internet Explorer browser window displaying a registration page for MarquetPlace.com. The browser's address bar contains the URL `http://localhost:8080/MP/WebRoot/user.jsp`. The page header features the site logo and navigation links: [About US](#), [Contact](#), [Site Map](#), [Link1](#), [Link2](#), and [Link3](#).

The main content area is divided into two columns. The left column contains a blue banner with the word "Développement", the text "N'hésitez pas...", and a promotional message: "Achetez des formations informatiques certifiées en ligne!". The right column contains a registration form titled "Inscription:" with the instruction "Veuillez saisir vos coordonnées ci-dessous:". The form fields are as follows:

Label	Value
Identifiant	idKader
Nom	Rahmani
Prénom	Kader
Adresse	Blida
E-mail	R_Kader@hotmail.com
Tel	025455523
Activité	Etudiant

An "Envoyer" button is located at the bottom right of the form. The browser's status bar at the bottom shows "Terminé" on the left and "Intranet local" on the right.

Figure V.12 Exemple d'identification d'un client

My JSP 'FormulaireClient.jsp' starting page - Microsoft Internet Explorer

Fichier Edition Affichage Favoris Outils ?

Précédente Recherche Favoris Média

Adresse Client.jsp?identifiant=myspass&nom=nserzr&pre nom=nserzr&adr=&email=&tel=&activite=&statu=&ok=Envoyer OK Liens

Search Mail My Yahoo! Sign In

l'agent délégué :

Identifiant

Type

Formation à acheter :

Domaine

Description

Date désiré

Prix min

Prix max

type

Terminé Intranet local

Figure V.13 Exemple d'identification des besoins d'un client

Après ces deux étapes, le client est bien inscrit au MPSMA et possède un agent mobile acheteur certifié de l'ASP. Après avoir cliqué sur le bouton Envoyer, L'agent acheteur délégué par ce client a lui ramené le résultat représenté par la figure suivante :

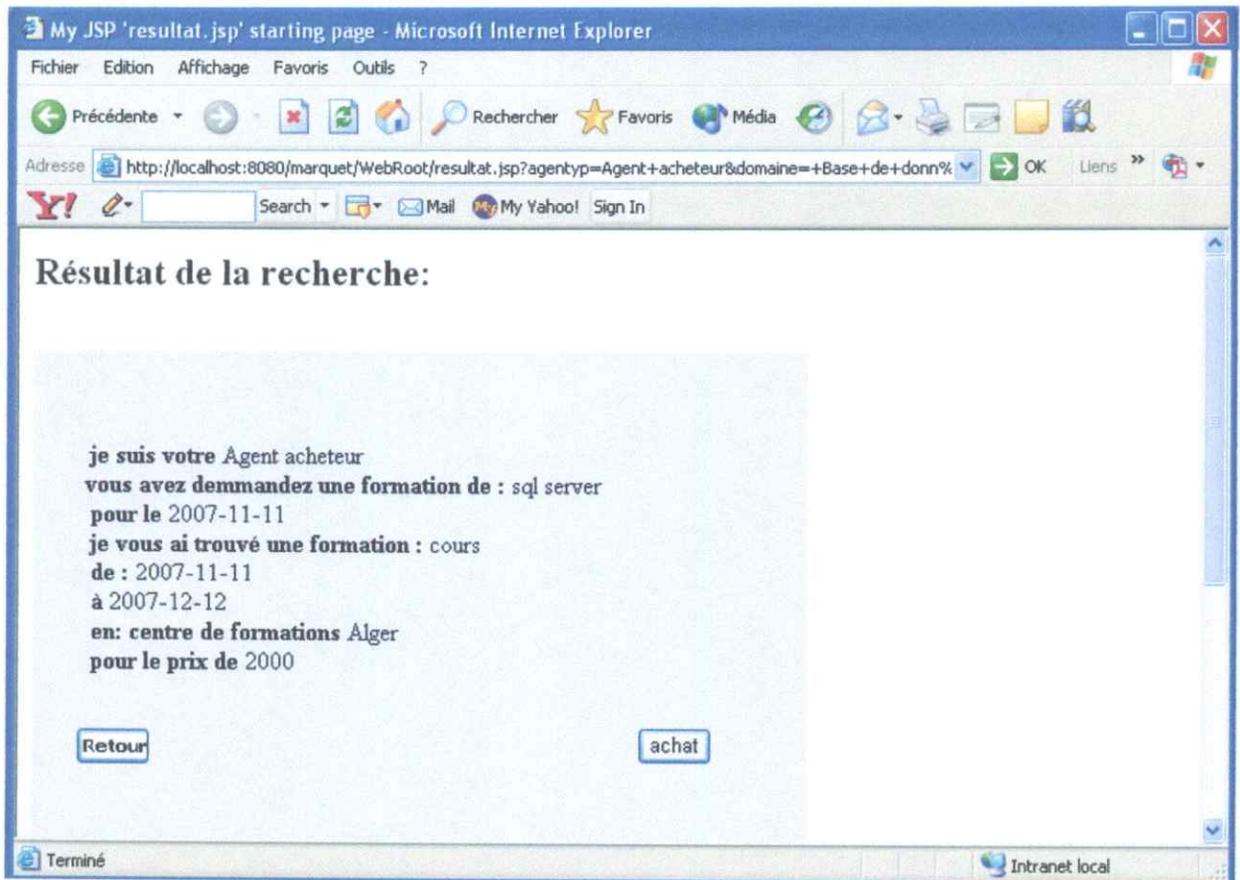


Figure V.14 Exemple d'un résultat apporté à un client

Si ce client soit satisfait par le résultat obtenu il affirmera son achat en cliquant sur le bouton achat ;

Remarque : dans le cas où son agent lui en a trouvé plusieurs selon ses critères de négociation (meilleur prix, date, et endroit), le choix de la formation sera une présélection ;

Après l'avoir fait, la forme de bon de commande illustrée en bas, sera rempli automatiquement, et lui apparaît pour la validation de la commande.

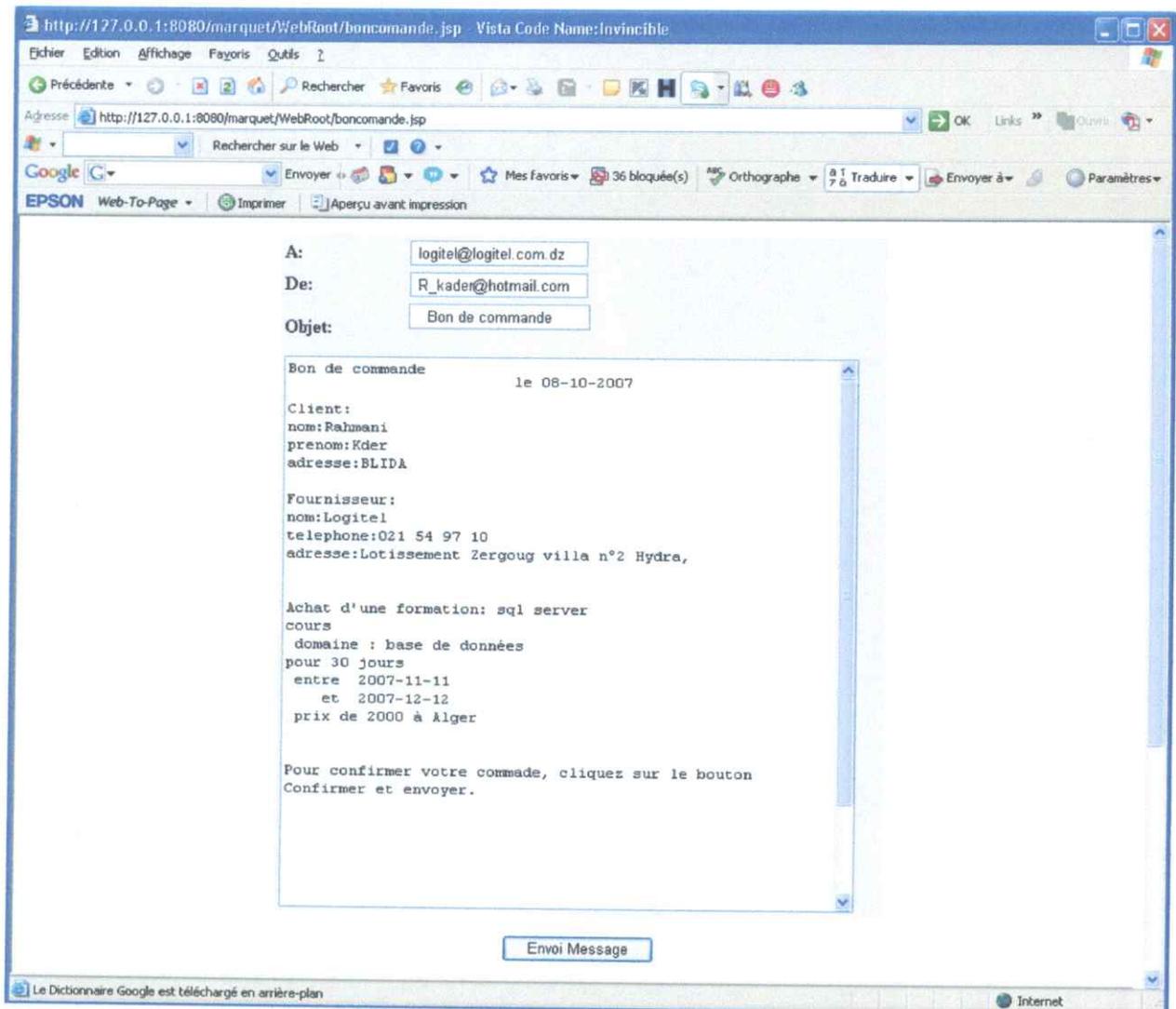


Figure V.15 Exemple d'un bon de commande

En cliquant sur ok l'achat sera bien effectué, et Kader va recevoir une accusé d'achat par son agent délégué.

Si le résultat de la transaction commerciale amené par l'agent acheteur de Kader est abouti, l'agent facilitateur du site MP vas mis en œuvre sa base MPDS, et l'actualiser de nouveau, et l'agent acheteur va être détruit.

Faisant la remarque, que dans une application similaire dans le cas de vendre une formation, l'agent vendeur qui la possède ne sera pas détruit, que, dans le cas de fin de son ttl ou bien si toutes ses offres sont vendus. À chaque fin d'une transaction commerciale quand l'ASP actualise son MPDS il revoit toute les ttl des agents vendeur

présenté dans la place de marché, et effectue les tâches correspondantes à chaque cas. Un agent est détruit en libérant son code.

Après avoir terminé la transaction effectuée par Kader l'ASP sauvegarde la trace dans sa base locale comme suit :

Tableau V.2 Exemple sur l'enregistrement de la trace d'une opération d'achat

numero	23
id_agent	idkader
typ_agent	acheteur
dat_creation	08/10/2007
certifié	1
dat_transaction	08/10/2007
result_transaction	réussite
dat_suppression	08/10/2007
Dat_fin_ttl	null
existe_actuellement	NON

6. CONCLUSION

Dans ce chapitre, nous avons présenté l'implémentation et la réalisation de notre système MPSMA.

Conclusion générale

Le but principal du présent travail consiste l'intégration du paradigme des agents et SMA (systèmes multi-agents) dans le commerce électronique. La motivation qui sous-tend ce travail réside principalement dans le fait que les agents offrent des tâches plus fiables, complexes et fastidieuses aux processus commerciaux : La spécification des besoins, La recherche (ou l'offre) des produits, La négociation, paiement en toute sécurité. Ce qui permet de rendre les places de marché (MP) plus efficaces avec une meilleure distribution des produits par rapport au besoins et gain du temps considérable.

Dans ce mémoire, nous avons présenté ce que sont les agents et les systèmes multiagents afin de mieux comprendre la dynamique de chacun d'entre eux. Nous y avons présenté les différentes architectures, et quelques plates-formes existantes, dans le but de mieux saisir le mode de fonctionnement d'un agent. Aussi, nous avons présenté différentes raisons qui justifient l'utilisation des systèmes multi-agents.

Nous avons également abordé un point essentiel caractérisant les SMA. Ce point concerne les interactions que les agents doivent effectuer à l'intérieur d'un système multiagents pour arriver à accomplir des tâches pour lesquelles ils sont destinés.

En revenant au cadre du commerce électronique, nous nous sommes intéressé aux trois architectures suivantes :

- ✓ **Kasbah** : un système multiagents où les utilisateurs qui veulent acheter ou vendre quelque chose, créent un agent, lui donnent des directions stratégiques de négociation, ils l'envoient ensuite sur un marché centralisé où cet agent va chercher un acheteur ou un vendeur potentiel (suivant le type de l'agent) avec qui il négociera.

- ✓ **MarketSpace** : Une infrastructure basée sur un paradigme agent. Le but de cette architecture est que le marché soit ouvert tout à fait comme le web.

- ✓ **MAGMA** : un prototype d'un système de marché virtuel. Il se compose d'un serveur de relais écrit en lisp et un ensemble d'agents écrits en Java. La version actuelle de Magma est conçue pour travailler avec les biens électroniques.

Finalement, et afin d'évaluer notre travail, nous l'avons mis en œuvre dans un système **MPSMA** basé sur **Java** et les outils Internet (**APACHE**, les pages **JSP**).

MPSMA est un système de commerce électronique sur Internet, conçu pour la commande et l'offre en ligne des formations informatiques.

Objectifs

Nous allons maintenant revoir l'essentiel de ce que nous avons accompli comme travail, à la lumière des objectifs que nous nous sommes fixés.

Objectif 1 : Etudier les mécanismes et le fonctionnement des agents intelligents

Nous avons présenté les notions d'agent, agent intelligent et les systèmes multi-agents, les diverses architectures et applications qu'ils les concernent.

Objectif 2 : Etudier les avantages des systèmes à agents mobiles en les comparant avec le système classique Client/serveur

Nous avons étudié les inconvénients des applications distribuées basées sur le système classique client/serveur tout en présentons les solutions amenées par les agents mobiles pour les éviter.

Objectif 3 : Présenter le commerce électronique et l'application des agents à ce domaine

Nous avons présenté les différents types de commerce électronique, l'application des agents dans le commerce électronique et des travaux réalisés dans ce cadre. Nous avons, par ailleurs, montré l'utilité des agents sur une place de marché électronique et discuté le choix des agents comme mécanisme adéquat pour les places de marchés.

Objectifs 4 : Etudier les architectures du commerce électronique

Nous avons présenté trois architectures **Kasbah**, **MarketSpace**, **Magma**. Ces dernières utilisent le paradigme agent dans leur conception.

Objectif 5 : la conception et la mise en œuvre du modèle suivi.

Nous avons décrit le modèle choisi en détail avec le langage UML, ensuite nous l'avons mis en œuvre avec JAVA et les outils Internet (APACHE, les pages JSP).

Perspective

Il nous reste comme perspective :

- ✓ Développer le processus de négociation entre les agents.
- ✓ Introduire le paiement bancaire dans la réalisation.
- ✓ Doter les agents mobiles de toutes les tâches d'achat et vente (les agents décident de mener la transaction du début à la fin)

A. HISTORIQUE

KQML a été développé durant les années 90 au sein du projet *Knowledge Sharing Effort* de la DARPA¹. Les travaux étaient dirigés par *Tom Finin* et *Jay Weber* et ses spécifications furent fournies en 1996.

A l'origine il fut conçu pour permettre l'interfaçage entre systèmes à base de connaissances. Il a rapidement été repensé pour devenir un langage de communication entre agent cognitifs. [35]

B. LES NIVEAUX DE KQML

KQML [PHPA 01] est structuré selon trois niveaux :

✓ *La couche de communication* : renseigne la communication (identité du récepteur, de l'émetteur et nature de la communication). Elle est minimale car KQML ne prend pas en charge le transport lui-même (TCP/IP, SMTP, IIOP ou autres).

✓ *La couche message* : donne des indications sur le contenu du message. C'est la couche centrale de KQML qui définit le type d'interaction que des agents-KQML pourront avoir.

✓ *La couche de contenu* : comporte la teneur réelle du message utilisant un langage de représentation propre au système. KQML peut supporter n'importe quel langage de représentation.

C. PRINCIPE DE KQML

Un message [36] KQML est caractérisé par sa performative², la classe du message, et un nombre variable d'attributs. La forme de base du protocole est :

(KQML-performative: sender <word>:receiver <word>:language <word> :ontology <word> : content <expression> ...)

- **language** (langage d'expression du message), **ontology** (le vocabulaire du domaine), **content** (le message). Les autres arguments d'un message sont :sender, :receiver, :reply-with, :in-reply-to.

Example: tell : sender Agt1 :receiver Agt2 :language KIF :ontology BlockWord :content (And (Block A) (Block B) (On A B)))

¹ DARPA¹, une agence américaine pour la recherche militaire

² Le terme *performatif* est utilisé pour identifier la force illocutoire d'une phrase.

D. Qualités de KQML :

Un tel modèle de [MARC 04] LCA présente certains avantages, ce qui faisait de lui, un langage fort utilisé.

- KQML étant, de fait, **le premier standard des LCAs**, cela lui confère certains avantages. Par exemple, plusieurs plate-formes multiagents n'ont eu d'autres choix que de l'utiliser et de nombreux analyseurs syntaxiques ont alors été développés pour répondre aux exigences.
- **Indépendance :**
 - KQML est indépendant du protocole de transport (ex. : TCP/IP, SMTP, IIOP).
 - KQML est indépendant du langage utilisé pour décrire le contenu (ex. : KIF, Prolog).
 - KQML est indépendant des ontologies que le contenu du message emploie.
- **Extensibilité :**
 - KQML permet l'ajout de performatifs.
 - KQML permet la création de nouvelles ontologies.

A. INTRODUCTION

Le premier algorithme de cryptage à clef publique a été développé par R. Merckle et M. Hellman en 1977. Il fut vite obsolète grâce aux travaux de Shamir, Zippel et Herlestman, de célèbres cryptanalyses.

En 1978, l'algorithme à clé publique de Rivest, Shamir et Adelman (R.S.A.) apparaît. Il sert encore au début du XXI^{ème} siècle à protéger les codes nucléaires des armées américaines et soviétiques.

Les procédés asymétriques sont, appelés « à clef publique » car ils sont conçus pour que l'une des deux clefs (la clef « publique ») puisse être révélée sans compromettre l'autre (la clef « secrète »). La clef secrète n'a jamais besoin d'être communiquée par son détenteur. La même clef peut par conséquent être employée pendant longtemps. [37]

RSA est [38] le plus célèbre et le plus répandu des algorithmes asymétriques

B. METHODE

Étant donnés p et q deux nombres premiers, $n=p*q$.

On choisit les entiers d et e tels que : e est premier avec $(p-1)*(q-1)$, $e < n$ et $(d*e-1)$ est multiple de $(p-1)*(q-1)$.

Le cryptage d'un nombre m inférieur à n est le nombre $c=m^e \bmod n$. On retrouve m par la formule soeur : $m=c^d \bmod n$.

Les couples (n, e) et (n, d) sont respectivement la clef publique et la clef privée.

C. FIABILITÉ

La donnée de n donne théoriquement accès à p et q qui sont simplement les facteurs premiers de n . Il se trouve cependant que la factorisation est réputée irréalisable, quand n est un très grand nombre (par exemple 2^{512}). Précisons toutefois que la difficulté de la factorisation n'est pas démontrée à ce jour. La complexité de cette opération n'est qu'un fait empirique. Quand on parle de la taille de la clef de RSA, il s'agit de la taille de n . On voit que c'est elle qui conditionne la fiabilité du cryptage.

En pratique, on utilise une clef publique (d) courte et une clef privée (e) longue, ce qui rend le décryptage plus rapide que le cryptage.

A. HISTORIQUE

UML³ est [MULL 01] un formalisme de modélisation orienté objet, résultat de l'unification des travaux de *James RUMBAUGH*, *Grady BOOCH* et *Ivar JACOBSON* auteurs respectivement, des méthodes *OMT*, *HOOD* et *OOSE*.

B. LES NOTIONS DE BASE

Le modèle conceptuel [MULL 01] d'UML est une abstraction d'un système développé en fonction de besoins ou d'objectifs. Il définit quelques notions de bases suivantes :

✓ **Les stéréotypes** sont des mécanismes d'extensibilités, ils sont utilisés pour définir une utilisation particulière d'éléments de modélisation existants ou pour modifier la signification d'un élément.

✓ **Les valeurs marquées** sont des paires (nom, valeur) qui ajoutent une nouvelle propriété à un élément de modélisation.

✓ **Les contraintes** sont des relations sémantiques entre les éléments de modélisation qui définissent des propositions devant être maintenues à vraies pour garantir la validité du système modélisé.

✓ **Les relations de dépendances** définissent des relations d'utilisations unidirectionnelles entre deux éléments de modélisation (source, cible) qui indiquent une situation dans laquelle un changement au niveau de la cible implique un changement au niveau de la source.

✓ **Les types primitifs** (Booléen, Entier, Expression, Nom, Chaîne, Temps) sont utilisés pour la modélisation d'UML dans le métamodèle.

✓ **Les paquetages** regroupent les éléments de modélisation : cas d'utilisation, classes, objets, modules ou composants. L'importation permet aux éléments d'un

³ UML : *Unified Modeling Language*

package d'accéder aux éléments d'un autre package. Cette relation est à sens unique et est représentée par une relation de dépendance associée à un stéréotype « import ».

C. LES DIAGRAMMES D'UML

Un diagramme d'UML [MULL 01] donne à l'utilisateur un moyen de visualiser et de manipuler des éléments de modélisation. UML définit des diagrammes structurels et comportementaux pour représenter respectivement des vues statiques et dynamiques d'un système.

C.1. Diagramme de classes

Les diagrammes de classes [MULL 01] expriment de manière générale la structure statique d'un système, en termes de classes et de relations entre ces classes.

C.2. Diagramme d'objets

Un diagramme d'objet [MULL 01] est une instance d'un diagramme de classes qui montre l'état du système modélisé à un instant donné.

C.3. Diagramme de composants

Les diagrammes de composants [MULL 01] décrivent les composants et leurs dépendances dans l'environnement de réalisation.

C.4. Diagramme de déploiement

Un diagramme de déploiement [MULL 01] décrit la disposition physique des ressources matérielles qui composent le système et montre la répartition des composants sur ces matériels. Chaque ressource étant matérialisée par un nœud, le diagramme de déploiement précise comment les composants sont répartis sur les nœuds et quelles sont les connexions entre les composants ou les nœuds.

C.5. Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation [MULL 01] décrivent le comportement d'un système au point de vue de l'utilisateur et fixe les limites du système et les relations entre le système et l'environnement.

C.6. Diagramme d'état-transition

Les diagrammes d'état-transition [MULL 01] décrivent le comportement interne d'un objet à l'aide d'un automate à états finis. Ils présentent les séquences possibles d'états et d'actions qu'une instance de classe peut traiter au cours de son cycle de vie en réaction à des événements discrets.

C.6. Diagramme d'activité

Un diagramme d'activité [MULL 01] est relativement proches des diagrammes d'états-transitions dans leur présentation, mais il est adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Il permette ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

C.7. Diagramme de collaboration

Un diagramme de collaboration [MULL 01] présente un ensemble de rôles joués par des objets et les liens entre ces objets. Ces liens montrent les interactions entre les objets à travers l'envoi de messages.

C.9. Diagramme de séquence

Un diagramme de séquence [MULL 01] montre les interactions entre les objets comme les diagrammes de collaboration. Toutefois il permet de modéliser la séquence des interactions et les scénarios selon un point de vue temporel.

Bibliographie

- [ACPM 98] Anthony Chavez, Pattie Maes , « KASBAH : An Agent Marketplace for Buying and selling Goods ». In 1st International Conference on Électronique Commerce 1998.
- [AEFS 05] Amal El Fallah Seghrouchni : «Introduction aux agents intelligents et aux systèmes multi-agents », Equipe SMA LIP6 - Université Paris 6,2005.
- [CHCC 05]Christophe CUBAT DIT CROS Agents Mobiles Coopérants pour les Environnements Dynamiques THESE présentée et soutenue publiquement le 2 décembre 2005 pour l'obtention du Doctorat de l'Institut National Polytechnique de Toulouse (ENSEEIH).
- [CHSI 98] Christian Simm : « Le commerce électronique, défis technologiques » , Swiss Science & Technologie Office for western USA and Wesern Canada,Février 1998.
- [CONC 00] Concordia Java Mobile Agent Technologie. <http://www.meitca.com/HSL/Projects/Concordia/2000>.
- [DANN 98] Danny B. Lange : « Present and future trends of mobile agent technology », Proceedings of the second International Workshop on Mobile Agents 98 (MA 98), Stuttgart, Germany, 9-11 Sept. 1998.
- [DGJP 99] David G. Messerschmitt, Jean Pierre Hubaux « Opportunities for E-commerce in Networking », Published in IEEE Communication Magazine, Sept.1999.
- [DJOH98] Dag Johansen : « Mobile agent applicability », 1998.
- [DJOH 04] Dag Johansen. Mobile agents : « Right concept, wrong approach. In Mobile Data Management», 2004.
- [DLMO 98] Danny B. Lange, MitsuruOshima, IBM research: «Programming and deployming Java Mobile Agents with Aglets », Addison-Wesley, 1998.
- [DOEW 97] R. Doorenbos, O.Etzioni, and D. Weld. «A Scable Comparaison-Shopping Agent for the World Wide Web. » Proceeding of the First International Conference on Autonomous Agents (Agent's 97). Marina del Rey, CA, February 1997. <http://www.jango.com>.
- [DRGC 91] Deborah Russell and G. T. Gangemi Sr: «Computer Security Basics» O'Reilly & Associates, Inc., 981 Chestnut Street, Newton, MA 02164, USA, 1991. A clear overview on many different security issues.
- [DTHA 02] Dillenseger B., Tagant A.-M., Hazard L: «Programming and Executing Telecommunication Service Logic with Moorea Reactive Mobile Agents», proc. 4th International

Workshop on Mobile Agents for Telecommunication Applications (MATA2002), Lecture Notes in Computer Science 2521, Springer, p. 48-57, Barcelona (Spain), October 2002.

[EFBF 04] El Falou S., Bourdon F. « Programmation répartie et agents mobiles », SETIT 2004.

[EFSJ 97] J. Eriksson, N.Finne and Sverker Janson. «MarketSpace : An Agent based market infrastructure » UPMAIL Technical Report No.147 April 7,1997 Box 311,S-751 05 Uppsala Sweden.

[FARA 03] Farah Abdel Majid BARIKA : « *Vers un IDS Intelligent à base d'Agents Mobiles* » Université de Tunis Institut Supérieur de Gestion Mémoire de DEA Pour obtenir Le Diplôme des Etudes Approfondies en Informatique Appliquée à la Gestion Année Universitaire 2002/2003.

[FIRE 00] Firefly (<[http:// www.firefly.com](http://www.firefly.com)>) 2000.

[FRAN 02] Frank Singhoff «La plate-forme Java RMI » Université de Brest Département/IUP informatique,2002.

[GCKR 01] R. S. Gray, G. Cybenko, D. Kotz, and D. Rus: «Mobile agents : Motivations and State of the Art». AAAI/MIT-Press, 2001.

[GRAY 96] R. Gray: «Agent Tcl: AFlexible and secure Mobile Agent System», Proceedings of the 4th Annual Tcl/Tk Workshop, Monterey CA, juillet 1996.

[HOFR 97] Hohl, Fritz: « An approach to solve the problem of malicious hosts». Universität Stuttgart, Fakultät Informatik, Fakultätsbericht Nr. 1997/03, 1997.

[JAME 95] James E. White: «Telescript technologie: « An introduction of the language», General Magic White paper GM-M-TSWP-0495-V1, General Magic.1995.

[JANG 01] Jango 2001 <http://www.jango.com>.

[JVRS 95] D.Johansen, R. Van Renssen et F.B. Scheinder: «An Introduction to the TACOMA Distributed System- version 1.0 »Technical Report 95-23, Universities of Tromso and Cornel, Juin 1995.

[JXHX 02] Lu Jun, Lu Xianlang, Han Hong, and Zhou Xu: « Application of mobile agent in wide area network». In IEEE, 2002.

[KRSG99] D. Kotz and R. S. Gray: «Mobile agents and the future of the internet ». Operating Systems Review, 33(3) :7-13, July 1999.

[KRUL 95] B. Krulwich. Bargain Fainder agent prototype. Technical report. Anderson Consulting 1995. [http:// bf.cstar.ac.com.bf](http://bf.cstar.ac.com.bf).

- [LANG 96] D. Lange: « Programming Mobile Agentin Java », Aglets White Paper, IBM Corp., <http://www.ibm.co.jp/tr/aglets.whitepaper.htm>, 1996.
- [LIDH 99] Leila Ismael and Daniel Hagimont: «A performance evaluation of mobile agent paradigm». In Proceedings of the International Conference on Object-Oriented Programming, Systems, Languages, and Applications (OOPSLA' 99), pages 306–313, Novembre 1999.
- [LILL 06] Thèse en vue de l'obtention du grade de DOCTEUR DE L'UNIVERSITE DES SCIENCES ET TECHNOLOGIES DE LILLE discipline «L'Interaction usagers/services, multimodale et multicanale , une première proposition appliquée au domaine du e-Commerce » 07 avril 2006.
- [LOUR 01] S. Loureiro. Mobile Code Protection. Thèse de doctorat, ENST Paris / Institut Eurecom, 2001.
- [MARC 03] Marc Saval : « Une architecture d'agents pour la simulation Le modèle YAMAM et sa plate-forme Phoenix » THÈSE présentée et soutenue publiquement le 27/06/2003 pour l'obtention du Doctorat de l'INSA de Rouen (Spécialité informatique).
- [MARC 04] Marc-André Labrie : « Langage de communication agent basé sur les engagements par l'entremise des jeux de dialogue » Mémoire présenté à la Faculté des études supérieures de l'Université Laval pour l'obtention du grade de maître ès sciences (M.Sc.) Département d'informatique et de génie logiciel, Janvier 2004.
- [MDWH 99] D. Milojicic, F. Douglis, and R. Wheeler: «Mobility : processes, computers and agents» Addison-Wesley, 1999.
- [MICH 98] Michel Paschalidés, « Le commerce électronique » 1998.
- [MNCR 04] D.E MENACER: « un modèle d'architecture à base d'agents mobiles pour les applications réparties » THESE pour obtenir le titre de Magister de l'institu National d'informatique-Alger, 2004
- [MULL 01] Alain Muller, modélisation objet avec UML, EYROLLES, 2001.
- [NICO 66] F. Nicosia. «Consumer Decision Processes : Marketing an Advertising Implications ». Prentice Hall, 1966.
- [NIES 02] Nicolas Esposito : « Coopeer : une architecture d'égal a égal pour la conception collaborative » Thèse pour obtenir le grade de Docteur de l'INPG préparée au laboratoire SIRAC (INRIA Rhône-Alpes). 09 septembre 2002
- [OBSP 97] ObjectSpace, Inc. ObjectSpace Voyager Core Package Overview. <http://www.objectspace.com>, 1997.

- [OLIV 02] Olivier Fourdrinoy « Création d'une librairie de Balises pour l'insertion d'agents JADE dans des pages Jsp ». Février , Mars, Mai , 2002 (stage informatique).
- [PERS 01] PersonaLogic (2001). <http://www.personalogic.com>.
- [PHPA 01] Philippe Pasquier « Communication entre agents », Doctorant DAMAS, université Laval, Canada, 13 septembre 2001. pasquier@iad.ift.ulaval.ca
- [PRES 99] présentation au Comité de travail du Groupe sur l'économie de l'information de l'OCDE « Business to Business E-Commerce Case Study: Transport and Logistics-Australia », document reprographié, 17 juin 1999.
- [RGPM 98] R. Guttman and P. Maes: «Agent-mediated Integrative Negotiation for Retail Electronic Commerce. » in the proceedings of the Workshop on Agent Mediated Electronic Trading (AMET'98), Minneapolis, Minnesota, April 1998.
- [RICH 98] R. Richard. « Guide du Commerce Electronique ». 1998.
- [SIRO 02] Siegfried Rouvrais. Utilisation d'Agents Mobiles pour la Construction de Services Distribués. Thèse, Université de Rennes 1, 2002.
- [STEP 97] Stéphane PERRET : « Agents mobiles pour l'accès nomade à l'information répartie dans les réseaux de grande envergure » THESE Pour obtenir le titre de Docteur de l'Université Joseph Fourier - Grenoble I, Novembre 1997.
- [SAGG 01] Stéphane Anglerot, Guillaume Bonnet, Guy Regnault « LES AGENTS INTELLIGENTS SUR INTERNET ». 03/10/2001.
- [TGMW 97] M. Tsvetovatyy, M. Gini, B. Mobasher , Z. Weickowski «MAGMA: An Agent-Based Virtuel Market for Electronic Commerce». Departement of Computer Science University of Minnesota, Minneapolis, USA 1997. <http://maya.cs.depaul.edu/~mobasher/papers/magma.pdf>.
- [YAKL 02] Youcef Aklouf, « VEMMA- une architecture de commerce électronique à base d'agent intelligent et mobile »thèse de magistère en informatique, USTHB d'alger, juin 2002 .

- [1] http://www.memoireonline.com/12/05/39/m_agents-intelligents.html
- [2] http://fr.wikipedia.org/wiki/Agent_intelligent#D.C3.A9finition
- [3] <http://www.daily.info/dossiers/multi-agents-eiah/index.php>
- [4] <http://www.grassouille.org/docs/memoire/node1.html>
- [5] http://opera.inrialpes.fr/people/Tayeb.Lemlouma/Papers/IAD_Presentation.pdf
- [6] <http://turing.cs.pub.ro/auf2/html/chapters/chapter1/sommaire.html>
- [7] http://www.limsi.fr/~jps/enseignement/examsma/2003/BACCAR_IBNELHEDJ/pagedescadres.htm
- [8] <http://www.tele.ucl.ac.be/EDU/ELEC2920/1997/agentsT/agents.html#1.Introduction>
- [9] [http://www2.ift.ulaval.ca/~kone/Cours/AM/AM_NatureProp.ppt#288,9.Nature\(2\)](http://www2.ift.ulaval.ca/~kone/Cours/AM/AM_NatureProp.ppt#288,9.Nature(2))
- [10] <http://www.limsi.fr/Individu/jps/enseignement/examsma/2004/LEGUERN/#0>
- [11] [http://www.agentintelligent.com/agent_intelligent/agents_intelligents.html#Caractéristiques des agents](http://www.agentintelligent.com/agent_intelligent/agents_intelligents.html#Caractéristiques_des_agents)
- [12] <http://www.limsi.fr/Individu/jps/enseignement/examsma/2004/LEGUERN/#0>
- [13] <http://www.tele.ucl.ac.be/EDU/ELEC2920/1997/agentsT/agents.html#3.3.1%20Architecture%20r>
- [14] [http://fr.wikipedia.org/wiki/Syst%C3%83%C2%A8mes_multiagents#Origine et aspects techniques](http://fr.wikipedia.org/wiki/Syst%C3%83%C2%A8mes_multiagents#Origine_et_aspects_techniques)
- [15] <http://www.limsi.fr/Individu/jps/enseignement/examsma/2002/nasriamrani/#Haut>
- [16] http://turing.cs.pub.ro/auf2/html/chapters/chapter1/chapter_1_2_4.html
- [17] [http://www.tele.ucl.ac.be/EDU/ELEC2920/1997/agentsT/agents.html#2.Domaines d'application](http://www.tele.ucl.ac.be/EDU/ELEC2920/1997/agentsT/agents.html#2.Domaines_d'application)
- [18] http://www.informatik.uni-stuttgart.de/cgibin/ncstr1_rep_view.pl?inf/ftp/pub/library/ncstr1.ustuttgart_fi/TR-1997-03/TR-1997-03.bib
- [19] http://jbenech.free.fr/old_site/agents/html/P-deja.html
- [20] http://noman.zerezo.com/PUBLIS/JavAne-IRIT_2003-25-R.pdf
- [21] [http://www.tele.ucl.ac.be/EDU/ELEC2920/1997/agentsT/agents.html#technologie Java](http://www.tele.ucl.ac.be/EDU/ELEC2920/1997/agentsT/agents.html#technologie_Java)
- [22] <http://webmaster.saliege.com/index.asp?page=news5/accueil>
- [23] <http://www.commentcamarche.net/entreprise/e-business.php3>
- [24] <http://www.allal-mennis.com>
- [25] <http://webmaster.saliege.com/index.asp>
- [26] <http://ditwww.epfl.ch/SIC/SA/publications/FI98/fi-2-98/2-98-page1.html>
- [27] <http://www.consommation.info.gouv.qc.ca/fr/fiche.asp?sujet=169>
- [28] http://www.cra-arc.gc.ca/notices/security_concerns-f.html#Top
- [29] <http://marketing-internet.com/articles/conception/index.html>
- [30] <http://laurent.caprani.free.fr/paiement/index.html#tdm>
- [31] http://cours2.fsa.ulaval.ca/cours/sio-65293/agents/e-commerce/agents_acheteurs.html
- [32] http://cours2.fsa.ulaval.ca/cours/sio-65293/agents/e-commerce/agents_vendeurs.html



- [33] <http://ditwww.epfl.ch/SIC/SA/publications/FI98/fi-2-98/2-98-page1.html>
- [34] <http://www.ulcc.ca/fr/cls/fkerr.pdf>
- [35] http://fr.wikipedia.org/w/index.php?title=Knowledge_Query_and_Manipulation_Language
- [36] http://www.loria.fr/equipes/maia/interne/maslect/chap2_resume.doc
- [37] <http://laurent.caprani.free.fr/paiement/techniques.html#cryptage>
- [38] <http://laurent.caprani.free.fr/paiement/bibliographie.html#RSA>